

UNIVERSIDAD NACIONAL DE PIURA

FACULTAD DE INGENIERÍA INDUSTRIAL

ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA

PROGRAMA DE ACTUALIZACIÓN PROFESIONAL

**PARA TITULACIÓN EN LA ESPECIALIDAD DE
INGENIERÍA INFORMÁTICA VERSIÓN XIX - 2019**



INFORME DE INVESTIGACIÓN

**“PROTOTIPO DE SISTEMA DE MONITOREO Y CONTROL DE
CONSUMO DE AGUA POTABLE APLICANDO TECNOLOGÍA
GSM”**

PRESENTADO POR:

Br. JARA NEYRA, KARINA

Br. ORDINOLA POICÓN, EVERT MANUEL

Br. SIANCAS PANTA, KAREN MABELY

**PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO INFORMÁTICO**

ASESORADO POR:

Ing. HEBERT EDUARDO ESPINO AGUIRRE, Mag.

LÍNEA DE INVESTIGACIÓN: INFORMÁTICA, ELECTRÓNICA Y
TELECOMUNICACIONES.

SUB LÍNEA DE INVESTIGACIÓN: PLATAFORMA DE TIC

Piura - Perú

2019

UNIVERSIDAD NACIONAL DE PIURA
FACULTAD DE INGENIERÍA INDUSTRIAL
ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA
PROGRAMA DE ACTUALIZACION PROFESIONAL
PARA TITULACIÓN EN LA ESPECIALIDAD DE
INGENIERÍA INFORMÁTICA VERSIÓN XIX - 2019



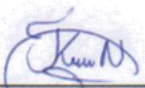
INFORME DE INVESTIGACIÓN

**“PROTOTIPO DE SISTEMA DE MONITOREO Y CONTROL DE
CONSUMO DE AGUA POTABLE APLICANDO TECNOLOGÍA
GSM”**


**PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO INFORMÁTICO**

**LÍNEA DE INVESTIGACIÓN: INFORMATICA, ELECTRONICA Y
TELECOMUNICACIONES.**

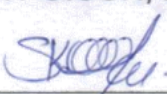
SUB LINEA DE INVESTIGACION: PLATAFORMA DE TIC



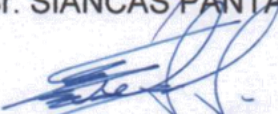
Br. JARA NEYRA, KARINA



Br. ORDINOLA POICÓN, EVERT MANUEL



Br. SIANCAS PANTA, KAREN



MG. ING ESPINO AGUIRRE, HEBERT EDUARDO
ASESOR

UNIVERSIDAD NACIONAL DE PIURA
FACULTAD DE INGENIERÍA INDUSTRIAL
ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA
PROGRAMA DE ACTUALIZACION PROFESIONAL
PARA TITULACIÓN EN LA ESPECIALIDAD DE
INGENIERÍA INFORMÁTICA VERSIÓN XIX - 2019



INFORME DE INVESTIGACIÓN

**“PROTOTIPO DE SISTEMA DE MONITOREO Y CONTROL DE
CONSUMO DE AGUA POTABLE APLICANDO TECNOLOGÍA
GSM”**

**PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO INFORMÁTICO**

**LÍNEA DE INVESTIGACIÓN: INFORMATICA, ELECTRONICA Y
TELECOMUNICACIONES.**

SUB LINEA DE INVESTIGACION: PLATAFORMA DE TIC

FIRMAS DEL JURADO:

MG. REYES VASQUEZ, DUBER
Miembro del Jurado Calificador

ING. CRUZ YARLEQUE, WILFREDO
Miembro del Jurado Calificador

DR. SAAVEDRA ARANGO, MOISES DAVID
Miembro del Jurado Calificador

DECLARACIÓN JURADA DE ORIGINALIDAD DEL INFORME DE INVESTIGACIÓN

Yo, KARINA JARA NEYRA, identificada con CU/ DNI N° 48045134, Bachiller de Escuela Profesional de INGENIERÍA INFORMÁTICA, de la Facultad de INGENIERIA INDUSTRIAL y domiciliada en A.H. JOSE CARLOS MARIATEGUI MANZANA B LOTE N° 15 CALLE ESTEBAN PABLETICH del Distrito de SULLANA, Provincia de SULLANA, Departamento PIURA.

Con Celular: 962644717. Email: karijn89@gmail.com

DECLARO BAJO JURAMENTO: que el Informe de Investigación que presento es original e inédita, no siendo copia parcial ni total de una tesis desarrollada, y/o realizada en el Perú o en el Extranjero, en caso contrario de resultar falsa la información que proporciono, me sujeto a los alcances de lo establecido en el Art. N°411, del código Penal concordante con el Art. N°32 de la Ley N° 27444, y Ley del Procedimiento Administrativo General y las Normas Legales de Protección a los Derechos de Autor.

En fe de lo cual firmo la presente.

Piura, 18 de julio del 2019



DNI N° 48045134

Artículo 411.- El que, en un procedimiento administrativo, hace una falsa declaración en relación con hechos o circunstancias que le corresponde probar, violando la presunción de veracidad por ley, será reprimido con pena privativa de libertad no menor de uno ni mayor de cuatro años.

Art. 4, Inciso 4.12 del Reglamento de Registro Nacional de Trabajos de Investigación para optar grados académicos y títulos profesionales -RENATI Resolución de Consejo Directivo N° 033-2013-SUNEDU/CD

DECLARACIÓN JURADA DE ORIGINALIDAD DEL INFORME DE INVESTIGACIÓN

Yo, EVERT MANUEL ORDINOLA POICÓN, identificado con CU/ DNI N° 46954568, Bachiller de Escuela Profesional de INGENIERÍA INFORMÁTICA, de la Facultad de INGENIERIA INDUSTRIAL y domiciliado en CALLE NUEVE N° 764 BARRIO BUENOS AIRES del Distrito de SULLANA, Provincia de SULLANA, Departamento PIURA.

Con Celular: 982609931. Email: ordinolapoiconmanuel@gmail.com

DECLARO BAJO JURAMENTO: que el Informe de Investigación que presento es original e inédita, no siendo copia parcial ni total de una tesis desarrollada, y/o realizada en el Perú o en el Extranjero, en caso contrario de resultar falsa la información que proporciono, me sujeto a los alcances de lo establecido en el Art. N°411, del código Penal concordante con el Art. N°32 de la Ley N° 27444, y Ley del Procedimiento Administrativo General y las Normas Legales de Protección a los Derechos de Autor.

En fe de lo cual firmo la presente.

• Piura, 18 de julio del 2019



DNI N° 46954568

Artículo 411.- El que, en un procedimiento administrativo, hace una falsa declaración en relación con hechos o circunstancias que le corresponde probar, violando la presunción de veracidad por ley, será reprimido con pena privativa de libertad no menor de uno ni mayor de cuatro años.

Art. 4, Inciso 4.12 del Reglamento de Registro Nacional de Trabajos de Investigación para optar grados académicos y títulos profesionales -RENATI Resolución de Consejo Directivo N° 033-2013-SUNEDU/CD

DECLARACIÓN JURADA DE ORIGINALIDAD DEL INFORME DE INVESTIGACIÓN

Yo, KAREN MABELY SIANCAS PANTA, identificada con DNI N° 44821182, Bachiller de Escuela Profesional de INGENIERÍA INFORMÁTICA, de la Facultad de INGENIERÍA INDUSTRIAL y domiciliada en A.H. VILLA PRIMAVERA MANZANA F1 LOTE N° 01 CALLE LOS FICUS del Distrito de SULLANA, Provincia de SULLANA, Departamento PIURA.

Con Celular: 948617441. Email: mabelk_01@hotmail.com

DECLARO BAJO JURAMENTO: que el Informe de Investigación que presento es original e inédita, no siendo copia parcial ni total de una tesis desarrollada, y/o realizada en el Perú o en el Extranjero, en caso contrario de resultar falsa la información que proporciono, me sujeto a los alcances de lo establecido en el Art. N°411, del código Penal concordante con el Art. N°32 de la Ley N° 27444, y Ley del Procedimiento Administrativo General y las Normas Legales de Protección a los Derechos de Autor.

En fe de lo cual firmo la presente.

Piura, 18 de julio de 2019



DNI N° 44821182

Artículo 411.- El que, en un procedimiento administrativo, hace una falsa declaración en relación con hechos o circunstancias que le corresponde probar, violando la presunción de veracidad por ley, será reprimido con pena privativa de libertad no menor de uno ni mayor de cuatro años.

Art. 4, Inciso 4.12 del Reglamento de Registro Nacional de Trabajos de Investigación para optar grados académicos y títulos profesionales -RENATI Resolución de Consejo Directivo N° 033-2013-SUNEDU/CD



UNIVERSIDAD NACIONAL DE PIURA
PROGRAMA DE ACTUALIZACIÓN PROFESIONAL
PATPRO EN INGENIERÍA INFORMÁTICA
VERSIÓN XIX - 2019



**ACTA DE EVALUACIÓN DEL INFORME DE
INVESTIGACIÓN**

Los Miembros del Jurado Calificador del Informe de Investigación denominado **"PROTOTIPO DE SISTEMA DE MONITOREO Y CONTROL DE CONSUMO DE AGUA POTABLE APLICANDO TECNOLOGIA GSM"**

presentado por los bachilleres: **ORDINOLA POICON EVERT MANUEL, JARA NEYRA KARINA Y SIANCAS PANTA KAREN MABELY**, participantes del Programa de actualización para Titulación Profesional en la Especialidad de Ingeniería Informática Versión XIX 2019; asesorados por el Mg. Hebert Espino Aguirre- Revisado y absueltas las observaciones formuladas por el Jurado Calificador, lo declaran:



Con la nota:

ORDINOLA POICON EVERT MANUEL

JARA NEYRA KARINA

SIANCAS PANTA KAREN MABELY

APROBADO

14

14

14

Piura, 13 de julio del 2019

11
Mg. DUBER REYES VASQUEZ
Miembro del Jurado

021
Ing. WILFREDO CRUZ YARLEQUE
Miembro del Jurado

Dr. MOISES DAVID SAAVEDRA ARANGO
Miembro del Jurado

DEDICATORIA

Con gran aprecio dedicamos este proyecto a Dios nuestro padre Celestial a quien le debemos todo lo que hemos logrado a lo largo de nuestras vidas

Y a nuestros padres quienes han sido nuestro gran apoyo y nuestra fuente de inspiración.

ÍNDICE GENERAL

INTRODUCCIÓN	3
CAPITULO I: ASPECTOS DE LA PROBLEMÁTICA	4
1.1. DESCRIPCIÓN DEL PROBLEMA.....	4
1.2. JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN.....	5
1.3. OBJETIVOS.....	5
1.3.1. Objetivo general.....	5
1.3.2. Objetivos específicos.....	5
1.4. DELIMITACIÓN DE LA INVESTIGACIÓN.....	6
1.4.1. Delimitación.....	6
CAPITULO II: FUNDAMENTO TEÓRICO.....	7
2.1. BASES TEÓRICAS.....	7
2.1.1. INTERNET DE LAS COSAS	7
2.1.2. ARDUINO	8
2.1.2.1. Definición.....	8
2.1.2.2. Arduino Uno	10
2.1.3. SENSORES Y ACTUADORES.....	13
2.1.4. SENSOR DE CAUDAL DE AGUA.....	14
2.1.5. GSM	14
2.1.6. GPRS	15
2.1.7. SIM900 GSM/GPRS.....	16
2.1.8. COMANDOS AT PARA PARA PROGRAMAR SIM900.....	17
2.1.9. ELECTROVÁLVULAS	18
2.1.10. MODULO RELÉ	19
2.1.11. AUTORIDAD NACIONAL DE AGUA.....	20
2.2. GLOSARIO	21
CAPITULO III: ASPECTOS ADMINISTRATIVOS.....	25
3.1. CRONOGRAMA DE EJECUCIÓN	25
3.2. PRESUPUESTO.....	26
CAPITULO IV: PROCEDIMIENTO Y DESCRIPCIÓN DEL PROTOTIPO	28
4.1. MATERIALES	28
4.2. PANEL FOTOGRÁFICO DE LOS MATERIALES.....	28
4.3. PROCEDIMIENTO Y DESCRIPCIÓN DEL PROTOTIPO.....	36
4.3.1. Calibración del sensor de caudal de agua YF-S201.....	36
4.3.2. Implementación de hardware.....	43

4.3.3. Implementación de software.....	48
4.3.4. Integración de hardware y software.....	57
4.3.5. Funcionamiento y resultados del prototipo.	61
4.3.5.1. Funcionamiento.....	61
4.3.5.2. Resultados.	68
CONCLUSIONES.....	74
RECOMENDACIONES	75
REFERENCIAS BIBLIOGRÁFICAS	76
ANEXOS	78

ÍNDICE DE FIGURAS

Figura 2. 1.- El IoT puede considerarse una red de redes	8
Figura 2. 2.- Arduino Uno R3.....	11
Figura 2. 3.- Sección de una Electroválvula.	19
Figura 4. 1.- Arduino Uno R3.....	29
Figura 4. 2.- Shield GSM/GPRS SIM 900 – Toma superior.	30
Figura 4. 3.- Shield GSM/GPRS SIM 900 - Toma inferior.	30
Figura 4. 4.- Sensor de caudal de agua YF-S201.	31
Figura 4. 5.- Electroválvula de 1/2" N/C 12V.....	32
Figura 4. 6.- Módulo Relay.	33
Figura 4. 7.- Protoboard de 830 puntos.....	33
Figura 4. 8.- Cargadores DC de 5V y 12v.	34
Figura 4. 9.- Cable Dupont.	34
Figura 4. 10.- Accesorios de Gasfitería	35
Figura 4. 11.- Caja de madera.....	35
Figura 4. 12.- Circuito a implementar.	36
Figura 4. 13.- Circuito para calibración de sensor.....	37
Figura 4. 14.- Sketch en Arduino.....	38
Figura 4. 15.- Pruebas de calibración.....	39
Figura 4. 16.- Prueba 1 de calibración – Volumen medio litro.	39
Figura 4. 17 Prueba 2 de calibración – Volumen medio litro.	40
Figura 4. 18.- Prueba 3 de calibración – Volumen medio litro.	40
Figura 4. 19.- Prueba 1 de calibración – Volumen un litro.....	41
Figura 4. 20.- Prueba 2 de calibración – Volumen un litro.....	41
Figura 4. 21.- Prueba 3 de calibración – Volumen un litro.....	42
Figura 4. 22.- Inserción del SIM Card al Shield GSM/GPRS SIM900.	43
Figura 4. 23.- Ubicación para Inserción del Shield GSM/GPRS SIM900 al Arduino.	43
Figura 4. 24.- Inserción del Shield GSM/GPRS SIM900 al Arduino.	44
Figura 4. 25.- Montado y conexión de Shield SIM900/Arduino al Protoboard.	44
Figura 4. 26.- Conexión del Relé.....	45
Figura 4. 27.- Conexión del sensor de caudal de agua.	46
Figura 4. 28.- Conexión de la electroválvula y cargador hacia el relé.	46

Figura 4. 29.- Montado de circuito a la caja y conexión de accesorios de gasfitería.	47
Figura 4. 30.- Hardware terminado.	47
Figura 4. 31.- Inicialización de variables.	50
Figura 4. 32.- Calcular N° de pulsos y Volumen.	51
Figura 4. 33.- Función para inicialización de variables.	52
Figura 4. 34.- Función loop - Envío de mensajes de texto.	53
Figura 4. 35.- Verificación de comandos AT.	54
Figura 4. 36.- Encendido y Apagado del SIM900.	55
Figura 4. 37.- Envío de comandos AT.	56
Figura 4. 38.- Envío de mensajes de texto.	57
Figura 4. 39.- Conexión para la integración del prototipo.	58
Figura 4. 40.- Subida del programa a la placa de Arduino.	59
Figura 4. 41.- Inicio del SIM900.	60
Figura 4. 42.- SIM900 Preparado.	60
Figura 4. 43.- Muestra de fluido de agua.	61
Figura 4. 44.- Envío de mensaje con la letra 'E'.	62
Figura 4. 45.- Recepción de mensaje 'LLAVE CERRADA'.	63
Figura 4. 46.- Corte de fluido de agua.	63
Figura 4. 47.- Envío de mensaje con la letra 'A'.	64
Figura 4. 48.- Recepción de mensaje 'LLAVE ABIERTA'.	65
Figura 4. 49.- Regreso de fluido de agua.	66
Figura 4. 50.- Envío de mensaje con la letra 'E'.	67
Figura 4. 51.- Recepción de mensaje con el consumo de agua.	68
Figura 4. 52.- Consumo de agua potable, sin controlar.	69
Figura 4. 53.- Consumo de agua potable controlado parte1.	71
Figura 4. 54.- Consumo de agua potable controlado parte2.	71

ÍNDICE DE TABLAS

Tabla 2. 1.- Características técnicas de Arduino Uno.	10
Tabla 3. 1.- Cronograma de ejecución.	25
Tabla 3. 2.- Presupuesto.	26
Tabla 3. 3.- Presupuesto de los materiales para el prototipo.	27
Tabla 4. 1.- Resumen de datos de la prueba de calibración.	42
Tabla 4. 2.- Resumen de datos de consumo de agua sin controlar.	69
Tabla 4. 3.- Resumen de datos de consumo de agua controlado.	72
Tabla 4. 4.- Cuadro comparativo de consumo de agua acumulativa.....	72

ÍNDICE DE GRÁFICOS

Gráfico 4. 1.- Consumo de agua por día sin controlar	70
Gráfico 4. 2.- Consumo de agua por día, controlado.....	72
Gráfico 4. 3.- Comparativo de agua potable acumulativo.....	73

RESUMEN

El presente proyecto está basado en una línea de investigación, del Internet de las Cosas (IoT) y surge tras el aumento de demanda de agua potable, producto del incremento de la población y el desperdicio de este recurso, hacen evidente la importancia del monitoreo y control del consumo del recurso más importante en nuestras vidas, aún hace falta seguir tomando conciencia de esa realidad ya que en el Perú se pierde el 37% del agua anualmente, teniendo en nuestras redes alrededor de 30,000 millones de metros cúbicos de agua, lo que significa, en teoría, un abastecimiento de 1000 metros cúbicos por habitante, pero el desperdicio es inmenso, según reveló el entonces Jefe de la Autoridad Nacional del Agua (ANA), Abelardo de la Torre (2018).

Basándonos en los datos del ANA, el desperdicio del agua es considerable sobre todo en nuestros hogares, dejando la llave abierta o teniendo una fuga de agua, o desperdiciando el agua en nuestros jardines, etc. Para ayudar a la conservación de este recurso se desarrolló un prototipo de monitoreo y control de agua potable aplicando el Sistema Global para las Comunicaciones Móviles (GSM), el cual permitió monitorear el consumo de agua en metros cúbicos, y controlar la apertura y cierre de una electroválvula a través de un mensaje de texto permitiendo el fluido del agua, se utilizaron como materiales, una tarjeta Arduino, un relé, una electroválvula, un módulo GSM y un sensor de flujo de agua, este último es un sensor de tipo propela, el cual permite medir el volumen del agua y llevar el control de lectura del consumo, cabe señalar que esta lectura es acumulativa.

Este prototipo se realizó con el fin de poder controlar de manera óptima el consumo de agua, evitar su desperdicio y coadyuvar a la conservación de este recurso de vital importancia en nuestra vida cotidiana, y así mismo saber el consumo mediante las lecturas.

Palabras claves: Desperdicio de agua, prototipo, Arduino, módulo GSM, sensor de flujo de agua y consumo de agua potable.

ABSTRACT

This project is based on a line of research, the Internet of Things (IoT) and arises after the increase in demand for drinking water, product of the increase in population and the waste of this resource, they make evident the importance of monitoring and We still need to continue to be aware of this reality, since in Peru 37% of the water is lost annually, with around 30 billion cubic meters of water in our networks. it means, in theory, a supply of 1000 cubic meters per inhabitant, but the waste is immense, according to the then head of the National Water Authority (ANA), Abelardo de la Torre (2018).

Basing us in the data of ANA, the waste of water is considerable in our Homes, because we leave the Open key or have leak and waste the water in our gardens, etc. For help to conserve this resource, we have developed a prototype of monitoring and control of potable water applying the Global System for mobile communications (GSM), this has allowed to monitor the consumption of water in cubic meters, and control the opening and closing of a electrovalve through a message of text, this allowed the water fluid. For this, we have used an Arduino card, a relay, a electrovalve, a GSM module, and a water flow sensor, this is a sensor of type propeller, that allow to measure the volume of water, and have a control of reading of consumption, is important to stand out that this reading is cumulative.

This prototype has developed with the purpose of carry out the control of optimal way of the consumption of water, prevent its waste, and help to the conservation of this vital resource in our daily life, and also know the consumption through the readings.

Keywords: Waste of water, prototype, Arduino, GSM module, water flow sensor and drinking water consumption.

INTRODUCCIÓN

Actualmente en nuestro país se viene concientizando a las personas de la importancia de agua, y las cifras del desperdicio de la misma son alarmantes, según datos revelados por la Autoridad Nacional del Agua (ANA), 10 millones de habitantes padecen de escasez o no tienen acceso al este recurso.

Para ayudar a la conservación y evitar al máximo su desperdicio se implementó un prototipo de sistema de monitoreo y control de consumo de agua potable aplicando tecnología GSM. Por mensaje de texto se puede controlar una electroválvula permitiendo la apertura y cierre de forma inmediata, para el fluido y corte del agua, como también conocer el consumo de agua en metros cúbicos.

En el Capítulo I se presentan los aspectos de la problemática, donde se reflejan los datos que nos incidieron a la implementación del prototipo, y que además comprende, la justificación e importancia, objetivos, delimitación del proyecto.

En el Capítulo II se define el marco teórico, donde se señalan las bases teóricas, conceptos y especificaciones técnicas tales como Arduino Uno R3, SIM900, sensor de caudal de agua, electroválvula y relé; además de un glosario.

En el Capítulo III se describe los aspectos administrativos como el cronograma de ejecución, y el presupuesto del proyecto;

En el Capítulo IV en el que se detalla el procedimiento y descripción del prototipo, seccionado en lista de materiales, panel fotográfico, el procedimiento de implementación del prototipo, las pruebas que se realizaron y resultados que se obtuvieron.

CAPITULO I: ASPECTOS DE LA PROBLEMÁTICA

1.1.DESCRIPCIÓN DEL PROBLEMA

Durante los últimos años se ha intentado concientizar a la población sobre la importancia del agua, según el Programa de Naciones Unidas para el Desarrollo (PNUD), en el artículo publicado en la página web oficial de la Autoridad Nacional de Agua www.ana.gob.pe, indica, que más de 1.100 millones de personas en el mundo carecen agua potable y 31 países padecen escasez de ella y dos de cada cinco personas no cuentan con instalaciones adecuadas de saneamiento

En el marco de las actividades por la Semana Nacional del Agua 2018, el entonces jefe de la ANA, Abelardo de la Torre advirtió que en el Perú se desperdicia anualmente 37% del agua disponible, teniendo en redes alrededor de 30,000 millones de metros cúbicos de agua, lo que significa, en teoría, un abastecimiento de 1000 metros cúbicos por habitante, pero el desperdicio es inmenso. En el país existen 10 millones de habitantes que no tienen agua o tienen poco acceso al líquido, la autoridad además detalló que se viene trabajando en reducir su despilfarro y en la promoción de un uso responsable del líquido; información publicada en la república.

Para ayudar a la conservación de este recurso, hemos creído conveniente realizar un prototipo que monitoree y controle el consumo de agua en el hogar, de tal forma que el usuario podrá conocer el consumo de este recurso en metros cúbicos, y además podrá controlar mediante un mensaje de texto la apertura y cierre de la llave principal, para evitar así el desperdicio de este recurso, cuando se deja las llaves de agua abierta y no solo servirá para el hogar sino también que se puede aplicar en colegios, instituciones empresas, etc.; pero en esta oportunidad este proyecto solo se aplicara en el hogar.

1.2. JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN.

La escases del agua en nuestro planeta es un tema de inquietud para todos los países del mundo, y el Perú no se encuentra excluido de esto, uno de los factores que causa la escases del agua es el mal uso y elevado desperdicio de ella, que genera una gran preocupación a los consumidores y a pesar de los mayores esfuerzos que se hacen para concientizar a la población del buen uso de este líquido, muchos no le dan la debida importancia, por eso, para coadyuvar a la conservación de este recurso vital para la vida, se hace necesario realizar el proyecto de monitoreo y control de agua potable que conlleva a mejorar el uso de este recurso hídrico, utilizando Tecnologías Informáticas y de Comunicaciones (TICs), este control se llevara a cabo enviando un mensaje de texto para la apertura y cierre de una electroválvula, como para el monitoreo; recibiendo como respuesta otro mensaje de texto con el consumo de agua en metros cúbicos.

Con este proyecto se quiere lograr que los miembros del hogar tomen conciencia del uso adecuado del agua, evitar el desperdicio, ayudar a que otros hogares que aún no cuentan con agua potable puedan llegar a tenerlo, y obtener el uso del agua potable de manera ecoeficiente, integrando el uso de dispositivos móviles para poder monitorear y controlar este recurso contribuyendo con un servicio social.

1.3. OBJETIVOS.

1.3.1. Objetivo general.

Implementar un prototipo de sistema de monitoreo y control de consumo de agua potable aplicando tecnología GSM

1.3.2. Objetivos específicos.

- Diseñar el hardware y software del prototipo de monitoreo y control de consumo de agua potable.
- Obtener datos de consumo de agua en tiempo real aplicando tecnologías GSM.

- Gestionar los datos de flujo de agua potable.

DELIMITACIÓN DE LA INVESTIGACIÓN.

1.4.1. Delimitación.

El desarrollo este proyecto tiene como finalidad reducir el desperdicio de agua potable, teniendo un uso ecoeficiente de este recurso en los predios; para realización de este proyecto tuvo como tiempo de duración de 4 meses y las pruebas con el prototipo fueron realizadas desde el 25 de junio al 01 de julio de 2019, en una vivienda de nivel socioeconómico medio, ubicada en la segunda etapa de la urbanización los Tallanes bloque 4B departamento 301, en donde viven dos personas adultas y una menor de edad.

Para objetivo de este estudio se contó con los recursos económicos para el diseño del prototipo, es decir para la adquisición de todos los materiales empleados tanto *hardware* y *software*, así como para los recursos financieros para solventar los diversos servicios que fueron necesarios.

Durante el proyecto se presentó un inconveniente que impidió el grabado en memoria en el Arduino Uno, por lo que se cada vez que se apaga, este reinicia los valores de consumo de agua, ya que solo tiene 1KB de almacenamiento.

CAPITULO II: FUNDAMENTO TEÓRICO

2.1. BASES TEÓRICAS

2.1.1. INTERNET DE LAS COSAS

El Internet de las cosas, se perfila como una de las principales tendencias que dan forma al desarrollo de las tecnologías de las comunicaciones. Es el paso del Internet utilizado para la interconexión de objetos físicos que se comunican entre sí, con los seres humanos, para ofrecer un determinado servicio. Se basa en tres pilares principales: ser identificables, para comunicarse, para interactuar (Miorandi et al. 2012).

Según la publicación de la Fundación de la Innovación Bankinter (2011), el IoT consiste en que las cosas tengan conexión a Internet en cualquier momento y lugar; en un sentido más técnico, consiste en la integración de sensores y dispositivos en objetos cotidianos que quedan conectados a Internet a través de redes fijas e inalámbricas. El hecho de que Internet esté presente al mismo tiempo en todas partes permite que la adopción masiva de esta tecnología sea más factible. Dado su tamaño y coste, los sensores son fácilmente integrables en hogares, entornos de trabajo y lugares públicos. De esta manera, cualquier objeto es susceptible de ser conectado en la red. Además, el IoT implica que todo objeto puede ser una fuente de datos. Esto está empezando a transformar la forma de hacer negocios, la organización del sector público y el día a día de millones de personas.

Según Evans (2011), el IoT se compone de un conjunto disperso de redes dispares diseñadas a medida. Los coches de hoy en día, por ejemplo, cuentan con diversas redes para controlar el funcionamiento del motor, las funciones de seguridad, los sistemas de comunicaciones, etc. Los edificios comerciales y residenciales también tienen varios sistemas de control para la calefacción, la ventilación y el aire acondicionado (HVAC); el servicio telefónico; la seguridad, y la iluminación. A medida que evoluciona el IoT, estas redes y muchas otras, se conectarán y contarán con mayores funciones de seguridad, análisis y gestión. Esto permitirá que el IoT pueda contribuir más y mejor a que las personas consigan sus objetivos.

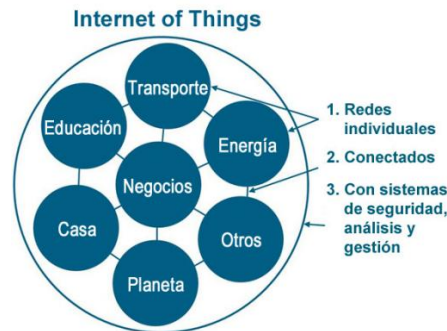


Figura 2. 1.- El IoT puede considerarse una red de redes
Fuente: Evans (2011)

2.1.2. ARDUINO

2.1.2.1. Definición

Según Indesa Company (2019); Arduino es una plataforma de prototipos electrónica de código abierto (open-source) basado en hardware y software flexible y fácil de usar. Está pensado para estudiantes, diseñadores, y para cualquiera interesado en crear objetos o entornos interactivos. Arduino puede interpretar el entorno mediante la recepción de sus entradas desde una variedad de sensores y puede afectar su entorno mediante el control de transductores, luces, motores y otros artefactos. El Microcontrolador de la placa se programa usando el “Arduino Programming Language” (basado en Wiring) y el “Arduino Development Environment” (basado en Processing). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador, tableta o equipo móvil. Las placas se pueden ensamblar a mano o encargarlas pre-ensambladas; el software se puede descargar gratuitamente. Los diseños de referencia del hardware están disponibles bajo licencia open-source, por lo que eres libre de adaptarlas a tus necesidades. Arduino es tres cosas:

- **Una placa hardware libre** que incorpora un microcontrolador reprogramable y una serie de pines

hembra (los cuales están unidos internamente a las patillas de E/S del microcontrolador) que permiten conectar allí de forma muy sencilla y cómoda diferentes sensores y actuadores. Cuando hablamos de "placa Arduino", deberíamos especificar el modelo concreto, ya que existen varias placas Arduino oficiales, cada una con diferentes características (como el tamaño físico, el número de pines-hembra ofrecidos, el modelo de microcontrolador incorporado y como consecuencia, entre otras cosas, la cantidad de memoria utilizable, etc.). Conviene conocer estas características para identificar qué placa Arduino es la que nos convendrá más en cada proyecto. (Artero, 2013)

- **Un Software** (más en concreto, un “entorno de desarrollo”) gratis, libre y multiplataforma (ya que funciona en Linux, MacOS y Windows) que debemos instalar en nuestro ordenador y que nos permite escribir, verificar y guardar ("cargar") en la memoria del microcontrolador de la placa Arduino el conjunto de instrucciones que deseamos que este empiece a ejecutar. El software o soporte lógico de un computador es el conjunto de programas asociados a un computador o a un equipo tecnológico que brinde funciones específicas estos pueden ser de muchos tipos de programación, de control de tratamiento, entre otros. (Álvarez, 2013).
- **Un lenguaje de programación libre ARDUINO (SDK):** El entorno del lenguaje de programación de Arduino, es fácil de usar para principales y lo suficiente flexible para los usuarios avanzados. Pensando en los profesores, Arduino está basado en el entorno de

programación de Processing con lo que el estudiante que aprenda a programar en este entorno se sentirá familiarizado con el entorno de desarrollo Arduino. (Torrente Artero, 2013).

2.1.2.2. Arduino Uno

Según Infootec (2019); Arduino Uno R3 es una placa electrónica de las muchas que tiene Arduino y con la que es muy fácil introducirse en el mundo de la programación electrónica, es una plataforma de código abierto (open-source) lo que permite realizar proyectos y modificaciones tanto de hardware como de software a cualquier persona sin ningún problema.

Tabla 2. 1.- Características técnicas de Arduino Uno.

CARACTERÍSTICAS TÉCNICAS	
Microcontrolador	ATmega328P
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7 - 12V
Voltaje de entrada (límite)	6 - 20V
Digital pines I/O	14
PWM digital pines I/O	6
Pines de entrada analógica	6
Corriente DC por Pin I/O	20mA
Corriente DC para Pin 3.3V	60mA
Memoria flash	32KB ATmega328P
SRAM	2KB ATmega328P
EEPROM	1KB ATmega328P
Velocidad de reloj	16 MHz
Longitud	68,6 mm
Anchura	53,4 mm
Peso	25g

Fuente: Página web del infootec.

La placa electrónica Arduino Uno R3 puede ser alimentada de varias formas, con un cable USB conectado al ordenador o con una fuente externa; cuenta con un zócalo donde se conecta un Jack de 2,1mm para conectar un adaptador que se encuentre entre los rangos de 7 – 12v que es la tensión

recomendada, así mismo cuenta con un conector USB tipo-B para conectarlo al ordenador con el cual se puede programar y a su vez alimentar.

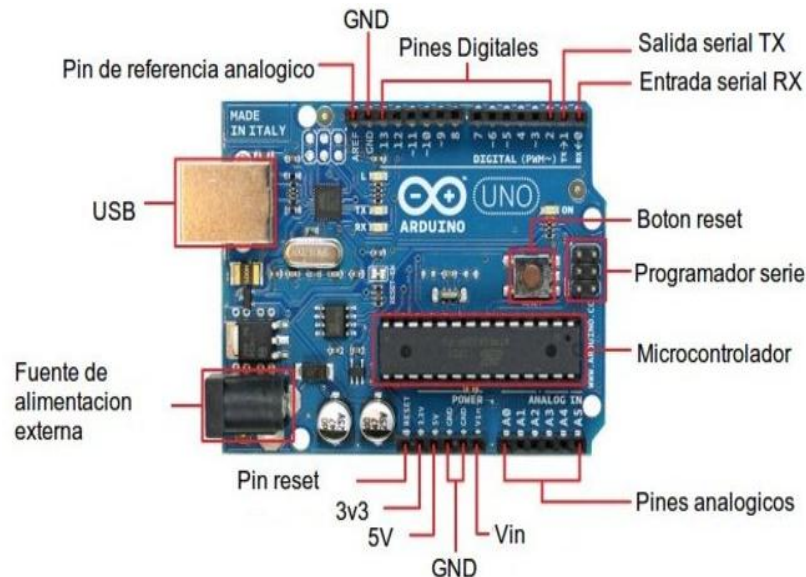


Figura 2. 2.- Arduino Uno R3.
Fuente: Página web del infofootet

Descripción de los pines de Arduino Uno

- Pin VIN: Este pin se puede usar de varias formas, si tenemos una fuente de alimentación conectada mediante un adaptador, lo que se puede hacer mediante este pin es obtener la alimentación para conectar otro dispositivo. Por otro lado, si se tiene conectado el USB, la tensión será regulada a 5v.
- Pin GND: El pin GND es la tierra.
- Pin 5v: Este pin tiene varias funciones, se puede alimentar la placa mediante este pin, siempre que se tenga la fuente externa regulada a 5v. Si se tiene la placa alimentada tanto por el Jack como por USB, se

puede alimentar otro componente con una tensión regulada de 5v.

- Pin 3.3v: Por este pin se tiene una tensión de 3.3v que es alimentada mediante el conector Jack o el USB. Los 3.3v se utilizan para alimentar dispositivos que requieren una tensión baja.

Pines de entradas analógicas:

La placa de Arduino cuenta con 6 pines de entradas analógicas, que van desde el pin A0 al A5, de los cuales proporcionan 10bits, llamados bits de resolución. La tensión que miden va de 0 a 5v, aunque es posible cambiar su rango usando una función con el pin AREF.

- Pin IOREF: El pin IOREF es una copia del pin VIN y se utiliza para indicar a los demás dispositivos conector a la placa que las tensiones de los pines de entrada y salida son 5v.
- Pin RESET: Este pin tiene el mismo funcionamiento que el botón RESET, se utiliza para reiniciar el microcontrolador.

Pines de entradas y salidas digitales:

Las entradas y salidas digitales son 14 y van desde el pin 0 al 13 y ofrecen una tensión de 5v.

- Pines A5 SCL y A4 SDA: Se pueden utilizar para conectar dispositivos que lleven a cabo comunicaciones mediante la librería Wire.

- Pin AREF: Ofrece un voltaje de referencia para las entradas analógicas.
- Pines 1 TX y 0 RX: Estos pines se utilizan para recibir y transmitir datos en serie.

2.1.3. SENSORES Y ACTUADORES

Entendemos por sensores a todos aquellos elementos que envían información de cómo se encuentran el automatismo o sobre los que podemos actuar, por ejemplo, pulsadores, finales de carrera, sensores, etc. Por otra parte, los actuadores serán los elementos finales de un automatismo: son los encargados de realizar las funciones de automatización: motores, cilindros neumáticos, resistencias, calefactores, etc., o incluso, lámparas de señalización. En la instalación de los automatismos estos elementos no van situados en el cuadro de control si no que se sitúan "a pie de máquina", como los finales de carrera, sensores de proximidad, termostatos, etc. (Páez, 2013). El sistema domótico está compuesto por tres elementos principales:

- **Sensores**, captan cualquier tipo de cambio físico en el interior de un ambiente y transmiten la información a la unidad de control para que actúe al modo de trabajo que se establecido.
- **Actuadores**, son aparatos que actúan con la unidad de control, transforma aquellos datos como: (subir persianas, realizar llamada, etc.).
- **Unidad de Control**, componente principal del sistema, es la parte encargada de gestionar la información y enviar los datos necesarios hacia el actuador para resolver los problemas. Tiene las interfaces necesarias para presentar la información por pantalla, monitor, etc.

2.1.4. SENSOR DE CAUDAL DE AGUA.

Según Edison Fabricio (2014); El sensor de flujo de agua se compone de una válvula con cuerpo de plástico, un rotor de agua, y un sensor de efecto Hall. Cuando el agua fluye a través de los rodillos del rotor el rotor gira, su velocidad cambia con diferente tasa de flujo, el sensor defecto hall da salida al impulso correspondiente a la señal, tiene una precisión aproximada del 30% de uno a 60 litros por minuto.

El efecto Hall es la medición del voltaje transversal en un conductor cuando es puesto en un campo magnético. Mediante esta medición es posible determinar el tipo, concentración y movilidad de portadores. Es decir, el sensor de efecto Hall crea un voltaje saliente proporcional al producto de la fuerza del campo magnético y de la corriente. Este tipo de medidores tienen un sensor que funciona a modo de switch ON/OFF. Cuando un imán pasa cerca del sensor se produce un voltaje en sus terminales, si el imán se retira el voltaje cae a 0V. Este imán está integrado en el rotor. Se muestra un sensor de la empresa SEED STUDIO que dispone de un cuerpo plástico, un rotor de agua, y un sensor de Efecto Hall. Cuando el agua fluye a través del rotor, el rotor rueda. Su velocidad cambia con diferente tasa de flujo (Viveros V, 2014).

2.1.5. GSM

Según informe consultado en la tesis: “Sistema de Control de Temperatura a través de Arduino y la Tecnología GPRS/GSM” Alberto Castro Domínguez (2000); GSM es la abreviatura de ‘Sistema Global para las comunicaciones Móviles’ (en inglés, Global System for Mobile communications). A comienzos del siglo XXI, es el estándar más utilizado de Europa. Conocido como estándar de segunda generación (2G), su principal diferencia respecto a la primera generación de teléfonos móviles es que sus comunicaciones son totalmente digitales. El estándar GSM fue desarrollado a partir

de 1982, cuando fue estandarizado por primera vez, denominado "Groupe Spécial Mobile". Surgió como idea para el desarrollo de un estándar europeo de telefonía móvil digital. En 1991 se convirtió en un estándar internacional llamado "Sistema Global de Comunicaciones Móviles", y comenzaron a presentarse los primeros prototipos de telefonía GSM. El estándar GSM permite transmisiones digitales de voz y datos, como mensajes de texto (SMS) o mensajes multimedia (MMS). Respecto a su arquitectura de red, en GSM todo terminal móvil debe estar constituido por una tarjeta SIM (Módulo de identificación de abonado) y el propio dispositivo, normalmente un teléfono móvil. La tarjeta SIM es la encargada de identificar en la red al usuario y al terminal móvil. Estos dispositivos se identifican gracias a un número -exclusivo de identificación denominado IMEI (Identificador internacional de equipos móviles), compuesto por 15 dígitos. Por otro lado, cada tarjeta SIM también posee un número de identificación único denominado IMSI (Identificador internacional de abonados móviles).

2.1.6. GPRS

Según informe consultado en la tesis: "Sistema de Control de Temperatura a través de Arduino y la Tecnología GPRS/GSM" Alberto Castro Domínguez (2000); define GPRS como GPRS o Servicio General de Paquetes vía Radio (en inglés, General Packet Radio Service) es una evolución del sistema GSM. Es también conocido como GSM++, pero dado que se trata de un estándar de telefonía móvil intermedio entre la segunda generación (2G) y la tercera (3G), a menudo recibe la nomenclatura de 2.5G. GPRS extiende la arquitectura del estándar GSM para permitir la transferencia de datos mediante conmutación de paquetes con velocidades de transferencia que rondan los 114 Kbps. Al contrario de lo que ocurre en conmutación de circuitos, en el estándar GPRS, gracias a su modo de transferencia de paquetes, las transmisiones de datos sólo utilizan la red cuando es necesario, permitiendo la

tarificación por volumen de información transmitida en lugar de por tiempo de conexión, por lo tanto, el usuario puede permanecer conectado sin costo adicional, ya que sólo utilizará la red cuando envíe o reciba un paquete de datos. Para el acceso a la red de datos, el estándar GPRS utiliza el protocolo IP, mientras que, para el transporte de voz, emplea la arquitectura de la red GSM. A parte de actualizar algunos servicios con los que ya contaba GSM, la tecnología GPRS admite otra serie de características que no estaban disponibles en 2G, Servicios de mensajes cortos (SMS) - Servicios de mensajes multimedia (MMS) - Servicio punto a punto (PTP); para la conexión cliente-servidor en una red IP, Servicio punto a multipunto (PTMP); para el envío de multidifusión.

2.1.7. SIM900 GSM/GPRS

Según Instructables Circuits (2019), indica que es una tarjeta GPRS ultra compacta de comunicación inalámbrica. La tarjeta es compatible con todos los modelos de Arduino con el formato UNO, además puedes controlarla con otros microcontroladores también. La tarjeta está basada en el módulo SIM900.

La tarjeta GPRS está configurada y controlada por vía UART usando comandos AT. Solo conecta la tarjeta al microcontrolador, Arduino, etc., y comienza a comunicarte a través de comandos AT. Ideal para sistemas remotos, comunicación recursiva, puntos de control, mandar mensajes de texto a celulares, etc. Especificaciones:

- Totalmente compatible con Arduino Conexión con el puerto serial
- Quad-Band 850/ 900/ 1800/ 1900 Mhz
- GPRS multi-slot clase 10/8GPRS mobile station clase B
- Compatible GSM fase 2/2+Clase 4 (2 W (AT) 850 / 900 MHz)
- Clase 1 (1 W (AT) 1800 / 1900MHz) TCP/UP embebido

- Soporta RTCConsumo de 1.5 mA (susp)

2.1.8. COMANDOS AT PARA PARA PROGRAMAR SIM900

Para programar las funcionalidades de la tarjeta se requiere emplear comandos AT y pese a que la lista de estos comandos es bastante extensa. Hay que aclarar que son los comandos AT y para ello, se puede decir que son lo que se conoce como el conjunto de comandos Hayes: un lenguaje desarrollado por la compañía Hayes Communications que prácticamente se convirtió en estándar abierto de comandos para configurar y parametrizar módems. Los caracteres «AT», que preceden a todos los comandos, significan «Atención», e hicieron que se conociera también a este conjunto de comandos como comandos AT.

Por un lado, aunque la lista de comandos AT es muy extensa, los comandos principales que se emplearán son los siguientes:

- **AT:** para verificar si el módulo SIM900 está funcionando adecuadamente para entrar en modo comando. Al enviar AT el SIM deberá contestarnos con un OK.
- **ATI:** Ver la información del producto.
- **AT+IPR=?:** el Baud Rate en el que puede operar el SIM.
- **AT+IPR?:** para preguntar el Baud Rate actual.
- **AT+IPR=XXXX:** configura la frecuencia deseada.
- **AT+COPS?:** nombre de la compañía telefónica.
- **AT+CGSN:** visualizar el IMEI del chip utilizado.
- **AT+CSCS?:** tipo de texto.
- **AT+CSCS="XXX":** Configurar a tipo de texto.
- **AT+CMGF?:** Ver el formato de un mensaje, ya sea PDU(0) o SMS(1)

- **AT+CMGF=1;** // modo texto para SMS
- **AT+CMGS=04455XXXXXXXXX:** Enviar un SMS Se despliega el símbolo mayor que ‘>’ Escribir mensaje y al finalizar presiona Ctrl+Z retornará OK si el SMS se envió correctamente.
- **AT+CMGL=ALL:** Sirve para ver todos los mensajes que nos han llegado al SIM.
- **ATD04455XXXXXXXXX:** Sirve para hacer una llamada a cualquier teléfono móvil.
- **ATA:** Sirve para contestar una llamada.
- **ATH:** Sirve para colgar una llamada.

(Alberto Lorenzo, 2004)

2.1.9. ELECTROVÁLVULAS

Las electroválvulas o válvulas electromagnéticas como son elementos mixtos, esto es, mediante una señal eléctrica exterior efectúan las funciones propias de las válvulas distribuidoras neumáticas. La parte fundamental de la electroválvula es el electroimán, capaz de mover directamente el distribuidor. Una electroválvula de mando indirecto funciona de la siguiente forma: Cuando la bobina está bajo tensión, el núcleo es atraído, venciendo el esfuerzo del muelle. En este momento, el conducto de alimentación neumático (P) puede alimentar el elemento neumático (A) porque los pequeños émbolos distribuidores, pilotados a través del conducto derivación de P, cierran la utilización (B) y abren la (A), respectivamente. Al cesar la tensión, el muelle devuelve al núcleo a su posición de reposo, se purgan los émbolos por la salida superior y el orificio (A) es puesto a escape (R) mientras (8) es alimentado. (Nistal, 2013).

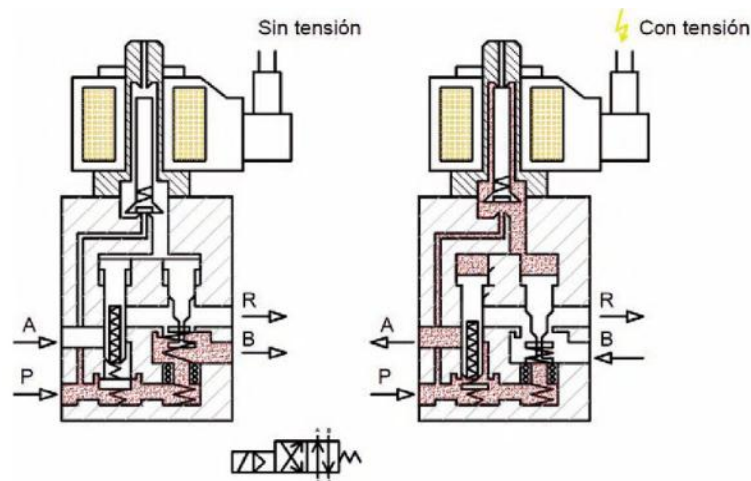


Figura 2. 3.- Sección de una Electroválvula.
Fuente: Nistal (2013)

2.1.10. MODULO RELÉ

Según Naylamp Mechatronics (2019); un módulo Relé permite controlar el encendido/apagado de equipos de alta potencia (electrodomésticos). Funciona perfectamente con Arduino, Pic o cualquier otro sistema digital.

El modulo posee 1 Relay de alta calidad, con capacidad de manejar cargas de hasta 250V/10A. El módulo relé posee un led indicador de alimentación (rojo) y un led indicador de activación (verde). Este módulo a diferencia de los módulos relé de 2 o más canales no posee optoacopladores, en su lugar la activación del relé es mediante un transistor. El diseño del módulo facilita el trabajo con Arduino, al igual que con muchos otros sistemas como Raspberry Pi, ESP8266 (NodeMCU y Wemos), Teensy y Pic. Este módulo Relay activa la salida normalmente abierta (NO: Normally Open) al recibir un "0" lógico (0 Voltios) y desactiva la salida con un "1" lógico (5 voltios). Para la programación del uso de Relays con Arduino se recomienda el uso de timers con la función "millis()" y de esa forma no utilizar la función "delay" que impide que el sistema continúe trabajando mientras se activa/desactiva un relé.

Entre las cargas que se pueden manejar tenemos: bombillas de luz, luminarias, motores AC (220V), motores DC, solenoides,

electroválvulas, calentadores de agua y una gran variedad de actuadores más. Se recomienda realizar y verificar las conexiones antes de alimentar el circuito, también es una buena práctica proteger el circuito dentro de un case.

2.1.11. AUTORIDAD NACIONAL DE AGUA

Según la Autoridad Nacional del Agua (2018), del Ministerio de Agricultura y Riego, de acuerdo a la Ley N° 29338 - Ley de Recursos Hídricos, es el ente rector y máxima autoridad técnico normativa del Sistema Nacional de Gestión de los Recursos Hídricos, el cual es parte del Sistema Nacional de Gestión Ambiental.

La ANA fue creada el 13 de marzo del 2008 por el Decreto Legislativo N°997, con el fin de administrar conservar, proteger y aprovechar los recursos hídricos de las diferentes cuencas de manera sostenible, promoviendo a su vez la cultura del agua, el actual Jefe es el Ing. Walter Obando Licera. La ANA es un organismo especializado, que pertenece al Ministerio de Agricultura y Riego (MINAGRI).

El ANA se encarga de:

- Administrar y vigilar las fuentes naturales de agua.
- Autorizar volúmenes de agua que utilizan y/o distribuyen los prestadores de servicios de agua (EPS y Juntas de regantes).
- Evaluar instrumentos ambientales.
- Otorgar derechos de uso de agua, autorizaciones de vertimiento y reúso de agua residual tratada.
- Autorizar obras en fuentes naturales de agua.
- Conducir el Sistema Nacional de Gestión de Recursos Hídricos.

La organización es de manera descentralizada a través de las 14 Autoridades Administrativas de Agua (AAA), 71 Administraciones

Locales de Agua (ALA) y 12 Consejos de Recursos Hídricos de Cuenca (CRHC). A través de ellas, se dirige y ejecuta el manejo de los recursos hídricos a nivel de cuencas de gestión; se aprueban estudios y obras de aprovechamiento de agua; se otorga derechos de uso de agua y autorizaciones de reúso de aguas residuales tratadas y de ejecución de obras; se vigila el uso de las fuentes de agua y se supervisa el cumplimiento del pago de retribución económica. Además, se realizan estudios, inventarios, monitoreos y la gestión de riesgos en glaciares, lagunas y fuentes de aguas subterráneas. Las ALAs apoyan a las AAA en sus funciones, entre ellas, capacitaciones, acciones de sensibilización y campañas de comunicación orientadas a promover la cultura del agua a nivel nacional.

2.2. GLOSARIO

Autoridad Nacional del Agua (ANA): Es el ente rector y máxima autoridad técnico – normativa del Sistema Nacional de Gestión de los Recursos Hídricos, adscrito al Ministerio de Agricultura y Riego, y está encargado de realizar las acciones necesarias para el aprovechamiento multisectorial y sostenible de los recursos hídricos por cuencas hidrográficas.

Autoridad Administrativa del Agua (AAA): Son órganos desconcentrados de la Autoridad Nacional del Agua dentro del país, que dirigen y ejecutan el manejo de los recursos hídricos a nivel de cuencas de gestión.

Autoridad Local del Agua (ALA): Dependen de la Autoridad Administrativa del Agua y se encuentran distribuidas en todo el país, en diversas cuencas hidrográficas; una de sus principales funciones es administrar los recursos hídricos a nivel de cuencas, entre otras

ARDUINO: Es una plataforma de prototipos electrónicos de código abierto.

AT: Este es el comando base de los comandos Hayes. Con él, comprobamos la disponibilidad del dispositivo, es una manera de conseguir comprobar que todo vaya bien

Consejos de Recursos Hídricos de Cuenca (CRHC): Son órganos de naturaleza permanente integrantes de la ANA, creados mediante D.S. a iniciativa de los Gobiernos Regionales, con el objeto de participar en la planificación, coordinación y concertación del aprovechamiento sostenible de los recursos hídricos en sus respectivos ámbitos.

DOMOTICA: Es el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda, que permite una gestión eficiente del uso de la energía, que aporta seguridad y confort, además de comunicación entre el usuario y el sistema.

ECOEficiencia: Es la gestión enfocada en utilizar menos recursos naturales y menos energía en los procesos productivos de una compañía, disminuir los residuos y desechos y atenuar la contaminación.

ÉMBOLO: Es la pieza móvil que se encuentra dentro del cuerpo de una bomba y que al moverse alternativamente cambia la presión de un fluido, generalmente con el objetivo de desplazarlo

GSM: Del inglés Global System for Mobile communications, en español Sistema Global para las comunicaciones Móviles. Es un sistema estándar, libre de regalías, de telefonía móvil digital.

GPRS: En inglés, General Packet Radio Service, es la abreviatura de Servicio General de Paquetes vía Radio. Es una extensión de la tecnología de comunicaciones móviles GSM, en ella la información es dividida en pequeños bloques, los que posteriormente se reagrupan al llegar a destino. Este tipo de transmisión permite una mayor capacidad y velocidad.

Internet de las cosas (IoT): Es un sistema de dispositivos de computación interrelacionados, máquinas mecánicas y digitales, objetos, animales o personas que tienen identificadores únicos y la capacidad de

transferir datos a través de una red, sin requerir de interacciones humano a humano o humano a computadora.

IMEI: En inglés se llama International Mobile Equipment Identity, en español se traduce como "identidad internacional de equipo móvil". Se trata de un número de 15 dígitos que lo identifica, está formado por cuatro apartados: los seis primeros dígitos, explican el país en el que se creó, los dos siguientes permiten saber quién es el fabricante, el siguiente es el número de serie y finalmente, un dígito verificador.

IMSI: Es el acrónimo de International Mobile Subscriber Identity (Identidad Internacional del Abonado Móvil). Es un código de identificación único para cada dispositivo de telefonía móvil, integrado en la tarjeta SIM, que permite su identificación a través de las redes GSM y UMTS.

OPTOACOPLADOR: Combina un dispositivo semiconductor formado por un fotoemisor, un fotoreceptor y entre ambos hay un camino por donde se transmite la luz, Todos estos elementos se encuentran dentro de un encapsulado que por lo general es del tipo DIP. son capaces de convertir una señal eléctrica en una señal luminosa modulada y volver a convertirla en una señal eléctrica. La gran ventaja reside en el aislamiento eléctrico que puede establecerse entre los circuitos de entrada y salida.

Programa de Naciones Unidas para el Desarrollo (PNUD): Su función es contribuir a la mejora de la calidad de vida de las naciones, promueve el cambio y centraliza el conocimiento, la experiencia y los recursos necesarios para ayudar a los pueblos a forjar una vida mejor.

SIM900: es una tarjeta GPRS ultra compacta de comunicación inalámbrica.

RELÉ: un interruptor controlado mediante un circuito eléctrico que, a través de un electroimán y de una bobina, define el cierre o a la apertura de otros circuitos.

UART: En inglés Universal Asynchronous Receiver Transmitter, en español: Transmisor Receptor Asíncrono Universal, es el dispositivo que

controla los puertos y dispositivos serie. Se encuentra integrado en la placa base o en la tarjeta adaptadora del dispositivo.

CAPITULO III: ASPECTOS ADMINISTRATIVOS

3.1. CRONOGRAMA DE EJECUCIÓN

Tabla 3. 1.-Cronograma de ejecución.

Actividad	Mes 1				Mes 2				Mes 3				Mes 4			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Recolección de información de fuentes primarias y secundarias																
Elaboración de los aspectos de la problemática y Marco teórico.																
Compra de materiales e implementación de Hardware																
Implementación de Software																
Funcionamiento de Prototipo y realización de pruebas																
Asesoría para evaluar los resultados.																
Formalización del informe final																

(Elaboración de los autores)

3.2. PRESUPUESTO.

- Presupuesto total.

Tabla 3. 2.- Presupuesto.

<u>SECCIONES</u>	<u>PARCIAL S/.</u>	<u>TOTAL S/.</u>
<u>Recurso Humano</u>		
Ejecutores	S/. 3 000.00	
Asesoría	S/. 3 600.00	
<u>Total Recurso Humano</u>	<u>S/. 6 600.00</u>	
<u>Bienes</u>		
Materiales para el prototipo	S/. 292.00	
1 Millar de Hoja DIN A-4	S/. 24.00	
3 CDs RW	S/. 10.00	
1 USB GB	S/. 32.00	
Lapiceros, resaltador.	S/. 10.00	
<u>Total Bienes</u>	<u>S/. 368.00</u>	
<u>Servicios</u>		
Internet	S/. 90.00	
Fotocopias	S/. 100.00	
Impresiones	S/. 200.00	
Movilidad	S/. 100.00	
<u>Total Servicios</u>	<u>S/. 490.00</u>	
<u>Total General</u>		<u>S/. 7 458.00</u>

(Elaboración de los autores)

El monto económico para realizar el proyecto es de S/7 458. 00 Nuevos soles, que será financiado por aporte de los autores.

- Presupuesto de los materiales para el prototipo.

Tabla 3. 3.- Presupuesto de los materiales para el prototipo.

<u>SECCION</u>	<u>PARCIAL S/.</u>	<u>TOTAL S/.</u>
<u>Materiales</u>		
Arduino Uno R3	S/. 38.00	
Un Shield GSM/GPRS SIM900.	S/. 105.00	
Sensor de caudal de agua YF-S201	S/. 22.00	
Electroválvula de ½" N/C 12 V	S/. 35.00	
SIM card	S/. 10.00	
Relé SRD-5VDC	S/. 10.00	
Protoboard	S/. 12.00	
Cables dupont.	S/. 5.00	
Accesorios de gasfitería	S/. 20.00	
Caja de madera	S/. 35.00	
<u>Total Materiales.</u>		S/. 292.00

(Elaboración de los autores)

CAPITULO IV: PROCEDIMIENTO Y DESCRIPCIÓN DEL PROTOTIPO

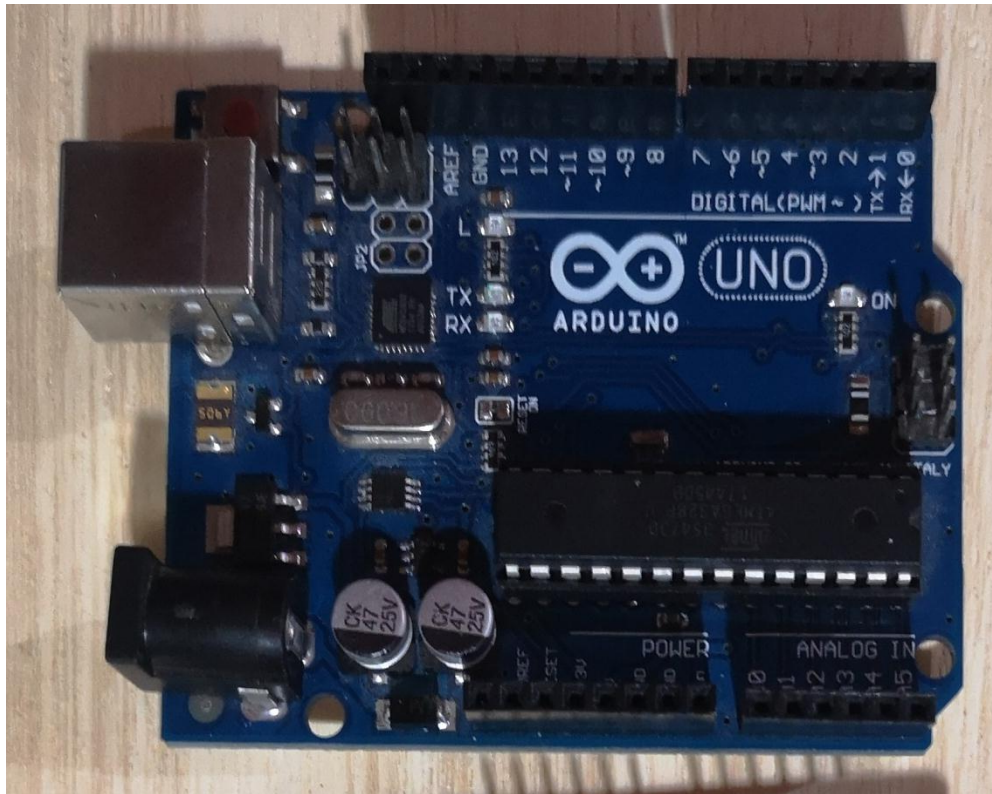
4.1. MATERIALES

Para el desarrollo del prototipo se necesitó la siguiente lista de materiales:

- Un Arduino UNO R3.
- Un Shield GSM/GPRS SIM900.
- Un Sensor de caudal de agua YF-S201 1/2" .
- Una Electroválvula de 1/2" N/C 12 V.
- Un Relé SRD-5VDC.
- SIM card.
- Un Protoboard de 830 puntos.
- Dos cargadores (5 V y 12 V)
- Cables dupont.
- Accesorios de gasfitería.
- Una caja de madera.

4.2. PANEL FOTOGRÁFICO DE LOS MATERIALES

- Arduino Uno R3; es la placa escogida para realizar este proyecto, por el hecho que es muy práctica de usar a comparación de los demás tipos de Arduino. La placa cuenta con un puerto USB que permite la conexión con un PC, para realizar la carga del programa y el intercambio de información. De igual modo, se ve que existen pines que permiten la transmisión serial con cualquier dispositivo que admita este sistema de comunicación; solo se van a utilizar 4 pines de entradas y/o salidas digitales.



**Figura 4. 1.- Arduino Uno R3.
(Captura fotográfica de los autores)**

- Shield GSM/GPRS SIM 90, será la encargada de permitir al sistema las comunicaciones, este necesitó un SIM card de cualquier operador para poder enviar y/o recibir los mensajes de texto. Se trata de una tarjeta o módulo de expansión perfectamente compatible con cualquier tarjeta del universo Arduino, incluso es posible encastrarla sobre la tarjeta Arduino UNO R3, debido a que los conectores que posee están alineados con esta finalidad.



Figura 4. 2.- Shield GSM/GPRS SIM 900 – Toma superior.
(Captura fotográfica de los autores)

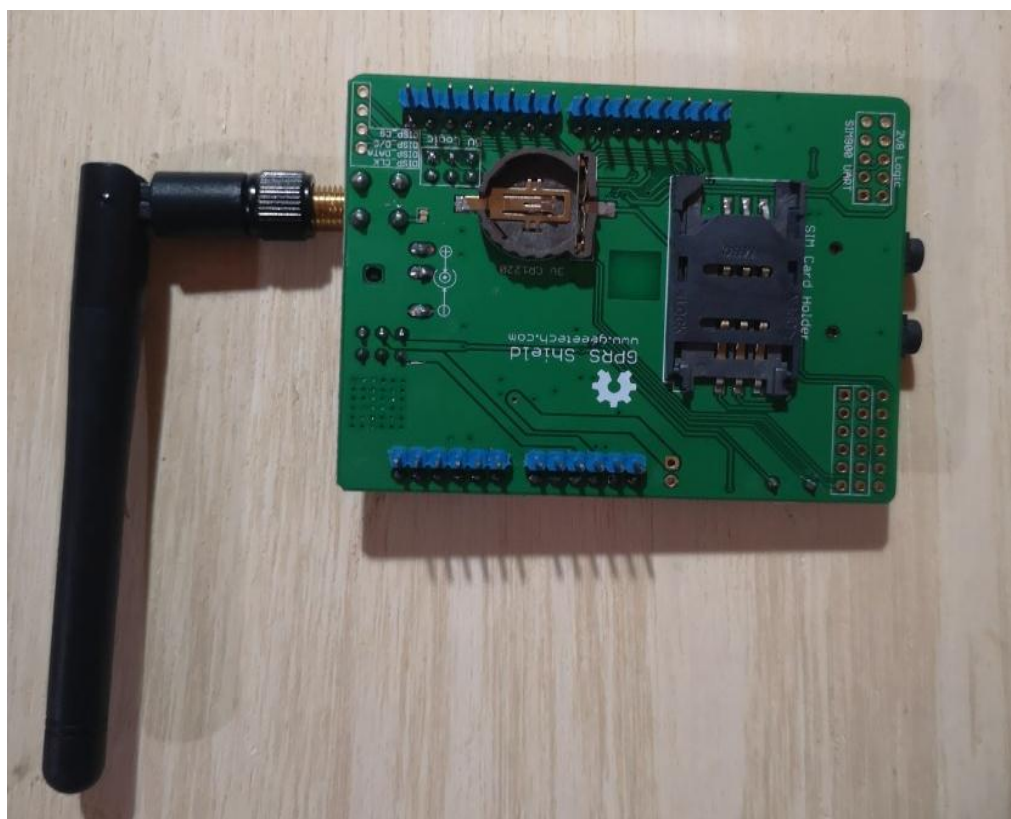


Figura 4. 3.- Shield GSM/GPRS SIM 900 - Toma inferior.
(Captura fotográfica de los autores)

- Sensor de flujo de agua YS-S201; es uno de los 4 sensores de caudal más utilizados con Arduino, y se escogió este tipo de sensor ya que el tamaño de la rosca de conexión es de ½ pulgada, indicado para conexiones de tubería domésticas. La conexión eléctrica es muy sencilla; todos los sensores de flujo tienen 3 cables: rojo y negro para positivo y GND; y amarillo para la salida de los pulsos; como fue necesario utilizar interrupciones externas, se utilizó el pin 2 de Arduino Uno, que es el que maneja interrupciones.



**Figura 4. 4.- Sensor de caudal de agua YF-S201.
(Captura fotográfica de los autores)**

- Electroválvula Solenoide de ½" N/C 12 V, es un tipo de electroválvula todo/nada o abierto/cerrado, Tienen dos partes: el solenoide y el cuerpo de plástico. El solenoide es un electroimán que al ser energizado se desplaza junto con el diafragma de la válvula y permite el paso del fluido; la válvula se mantiene abierta

mientras el solenoide está energizado; Cuando no está alimentado un resorte se encarga de regresar la válvula a su posición de reposo, que en este caso es del tipo normalmente cerrada (NC). Las tuberías de los hogares tienen un diámetro de ½ pulgada, ideal para controlar el flujo de agua en un hogar.



**Figura 4. 5.- Electroválvula de 1/2" N/C 12V.
(Captura fotográfica de los autores)**

- Modulo Relé SRD-5VDC; permite controlar el encendido/apagado de equipos de alta potencia; en el proyecto controla a la electroválvula de 12V, funciona perfectamente con Arduino, que trabaja a 5V, el modulo posee 1 Relé de alta calidad, con capacidad de manejar cargas de hasta 250V/10A. Este módulo relé activa la salida normalmente abierta al recibir un "0" lógico (0 Voltios) y desactiva la salida con un "1" lógico (5 voltios).



Figura 4. 6.- Módulo Relay.
(Captura fotográfica de los autores)

- Protoboard; es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, fue ideal para el circuito del prototipo.

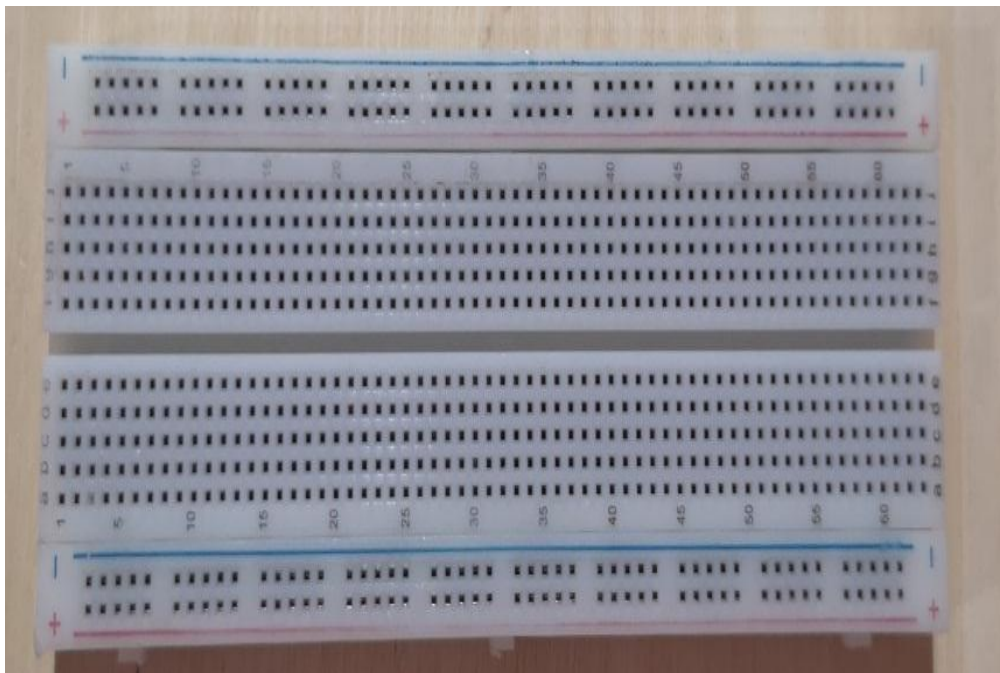


Figura 4. 7.- Protoboard de 830 puntos.
(Captura fotográfica de los autores)

- Se utilizaron dos cargadores uno de 5 voltios que va conectado al Arduino (trabaja a 5V) y uno de 12 voltios para la electroválvula (trabaja a 12 V).



**Figura 4. 8.- Cargadores DC de 5V y 12v.
(Captura fotográfica de los autores)**

- Cables dupont; son cables para realizar las conexiones en el protoboard, los que utilizaron fueron macho-macho.



**Figura 4. 9.- Cable Dupont.
(Captura fotográfica de los autores)**

- Accesorios de gasfitería, se utilizaron para realizar las conexiones



**Figura 4. 10.- Accesorios de Gasfitería
(Captura fotográfica de los autores)**

- Caja de madera para montar el prototipo.



**Figura 4. 11.- Caja de madera.
(Captura fotográfica de los autores)**

4.3.PROCEDIMIENTO Y DESCRIPCIÓN DEL PROTOTIPO.

El proyecto se constituyó con la plataforma de desarrollo Arduino y hardware (sensor, electroválvula, Arduino, dispositivo móvil, etc) antes descritos. Bajo estas tecnologías se da solución al problema que se planteó para evitar el desperdicio de agua potable y de la misma manera monitorear de consumo de agua. Para la implementación del circuito como muestra la figura 4.12 se siguió los siguientes pasos:

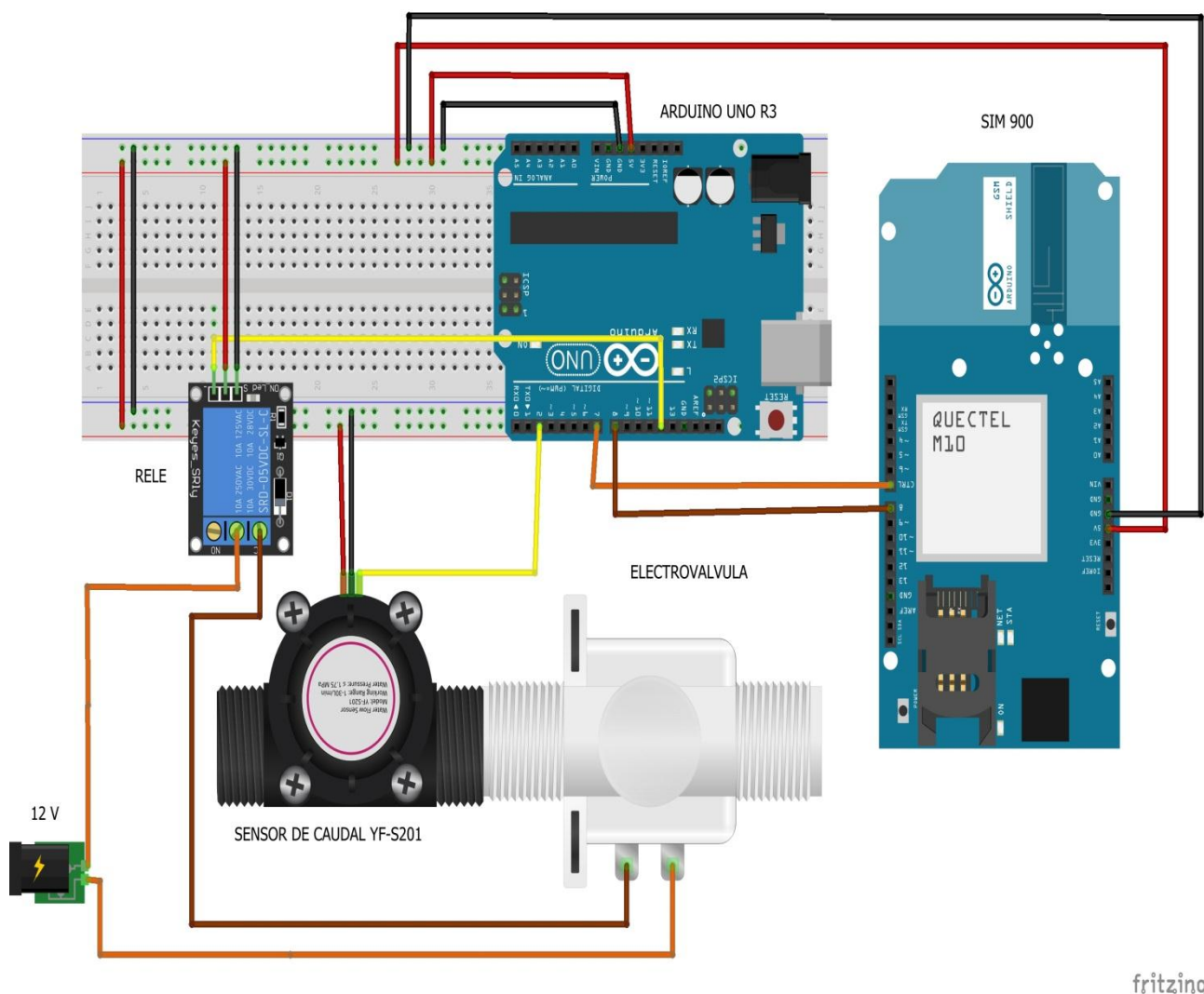


Figura 4. 12.- Circuito a implementar.
(Elaboración de los autores)

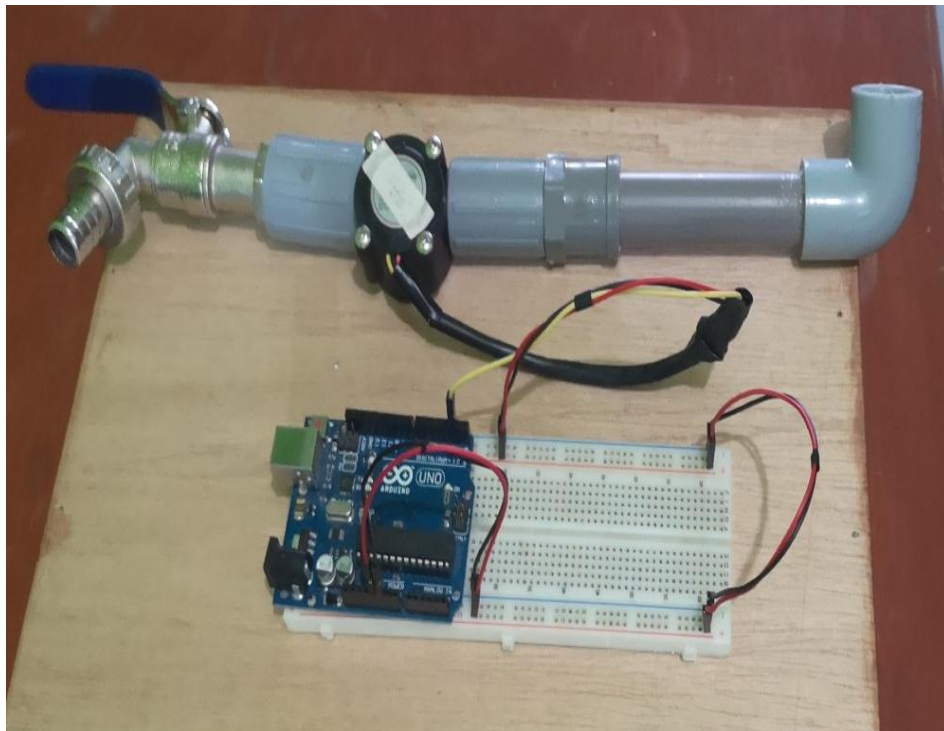
4.3.1. Calibración del sensor de caudal de agua YF-S201.

Para poder tener una medición exacta del flujo de agua se tuvo que calcular el valor del factor de conversión que es el que nos sirve para

pasar de frecuencia de pulsos a caudal de agua. Para ello se tuvo que implementar el siguiente circuito donde:

- Cable amarillo del sensor va conectado directamente al pin 2 del Arduino, utilizamos este pin porque en el programa vamos a usar la interrupción externa y Arduino Uno solo tiene interrupciones externas en los pines 2.
- Cable rojo al pin 5V.
- Cable negro a GND.

Como muestra la figura 4.13.



**Figura 4. 13.- Circuito para calibración de sensor.
(Captura fotográfica de los autores)**

La variable que se midió con exactitud es la cantidad de pulsos (usando el Arduino), y con ayuda de un recipiente con graduación se midió la cantidad o volumen de agua; para calcular el factor de conversión con estas dos variables se utilizó la siguiente fórmula

$$K = \frac{n^{\circ} \text{ pulsos}}{\text{volumen} \cdot 60}$$

Dónde:

- K, es el factor de conversión.
- n° pulsos, son la cantidad de pulsos medidos con el sensor, correspondientes al volumen que pasa por el mismo.
- V, es el volumen que se midió con ayuda de recipientes con graduación.

Se utilizó el siguiente Sketch, hecho en Arduino Versión 1.8.8, como muestra la figura 4.14.



```
CalibracionDeSensor

/*CALIBRACIÓN DEL SENSOR DE CAUDAL DE AGUA YF-201*/
volatile int pulsos; // Cantidad de pulsos del sensor. Como se usa dentro de una interrupcion debe ser volatile
#define sensorDeFlujo 2 //Pin al que se conecta el sensor. DEBE ser 2 porque es unico que acepta interrpciones en el Arduino UNO
void flujo () // Funcion de interrupcion
{
    pulsos++; // Simplemente sumar el numero de pulsos
}

void setup()

{
    pinMode(sensorDeFlujo, INPUT);
    Serial.begin(9600);
    attachInterrupt(0, flujo, RISING);
    interrupts();
}

void loop ()
{
    Serial.print("Pulsos: ");
    Serial.println(pulsos);
}
```

**Figura 4. 14.- Sketch en Arduino.
(Captura de pantalla)**

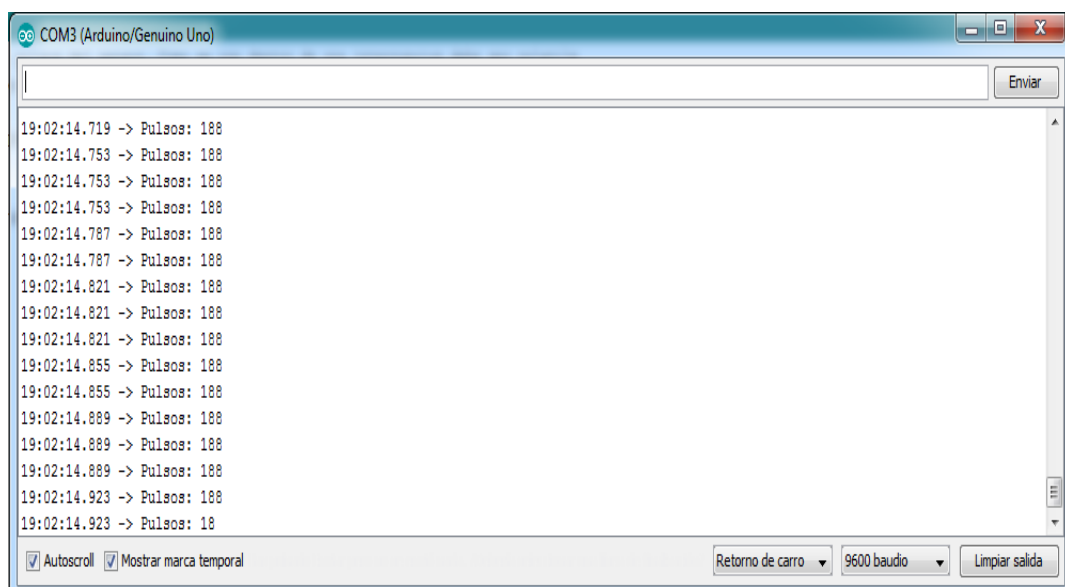
Se realizaron tres pruebas de medio litro y tres pruebas de un litro, como muestra la figura 4.15.



**Figura 4. 15.- Pruebas de calibración.
(Captura fotográfica de los autores)**

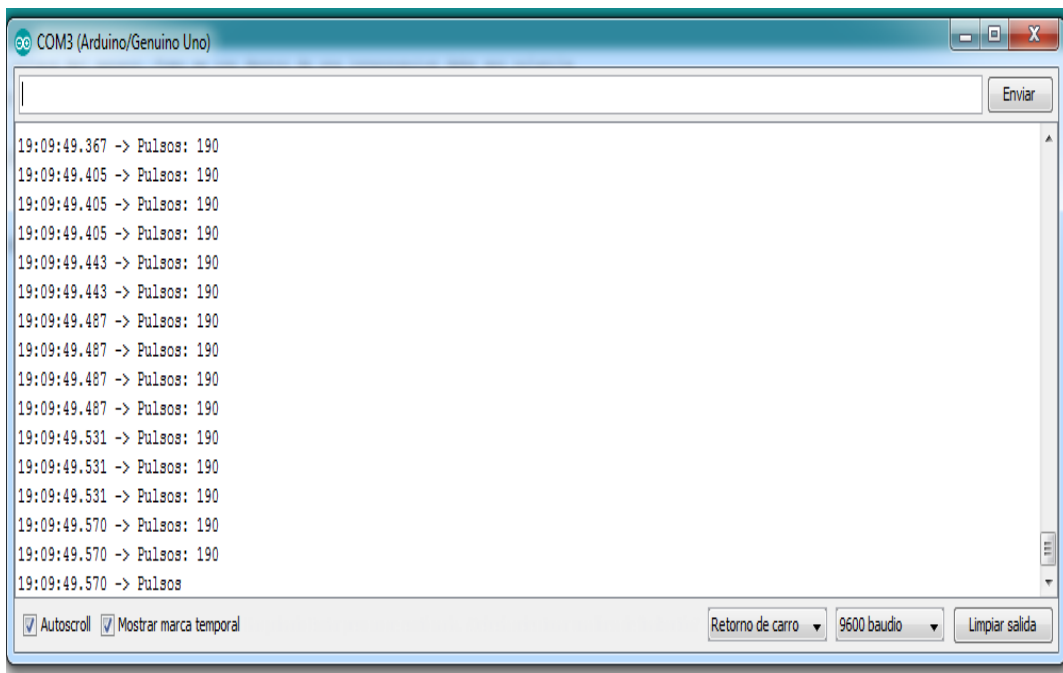
Se abrió el monitor serie y se realizó las tres primeras pruebas con el volumen de medio litro por cada una, obteniendo los siguientes resultados:

En la prueba 1 se obtuvo 188 pulsos, como muestra la figura 4.16.



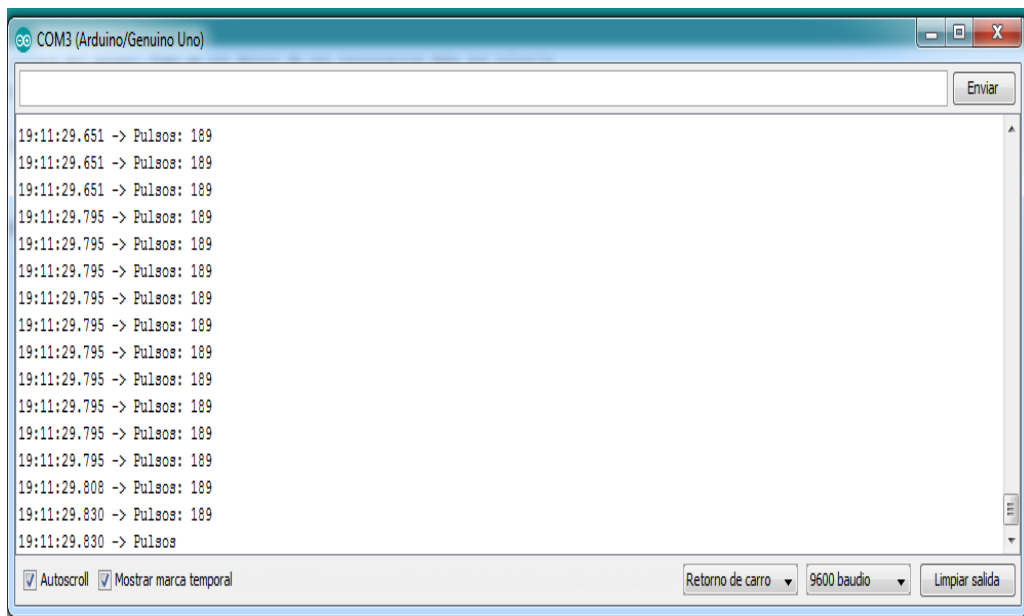
**Figura 4. 16.- Prueba 1 de calibración – Volumen medio litro.
(Captura de pantalla)**

En la prueba 2 se obtuvo 190 pulsos como muestra la figura 4.17.



**Figura 4. 17 Prueba 2 de calibración – Volumen medio litro.
(Captura de pantalla)**

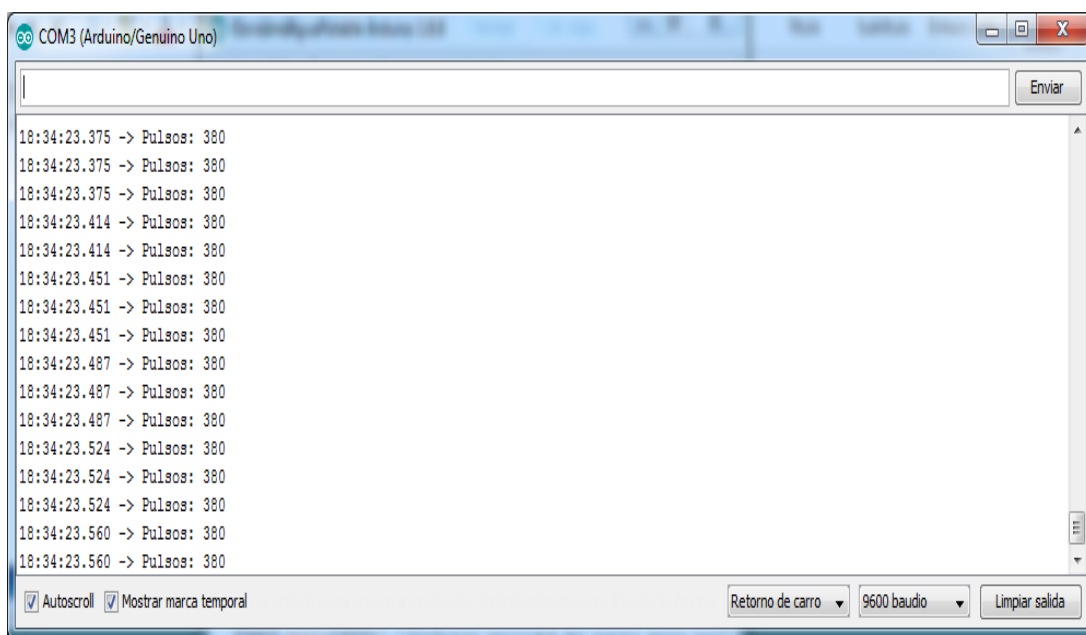
En la prueba 3 se obtuvo 189 pulsos, como muestra a figura 4.18.



**Figura 4. 18.- Prueba 3 de calibración – Volumen medio litro.
(Captura de pantalla)**

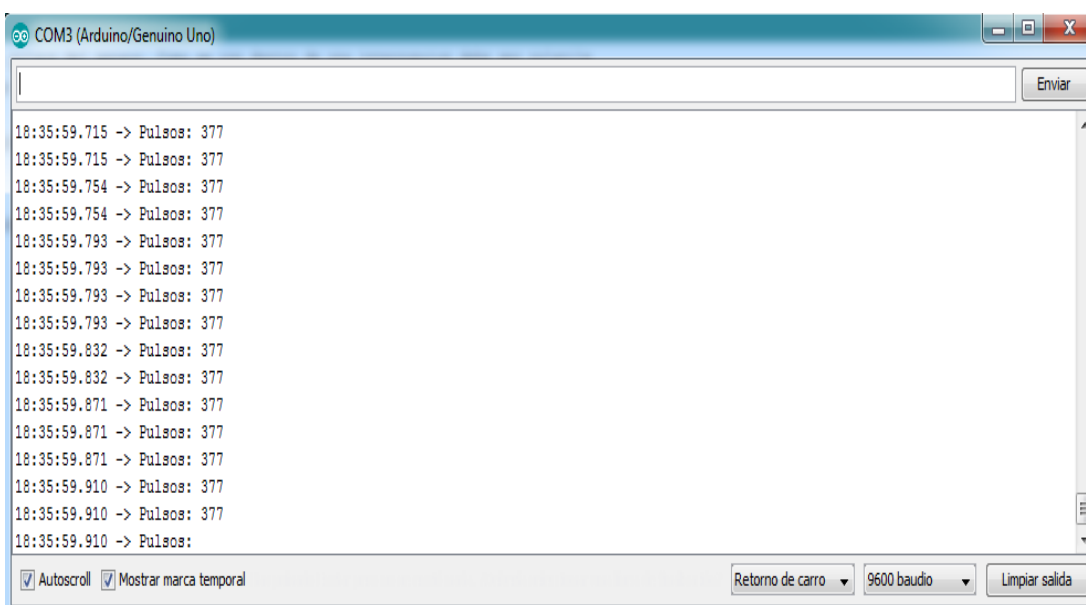
Luego se realizó las siguientes tres primeras pruebas con el volumen de un litro por cada una, obteniendo los siguientes resultados:

En la prueba 1 se obtuvo 380 pulsos, como muestra la figura 4.19:



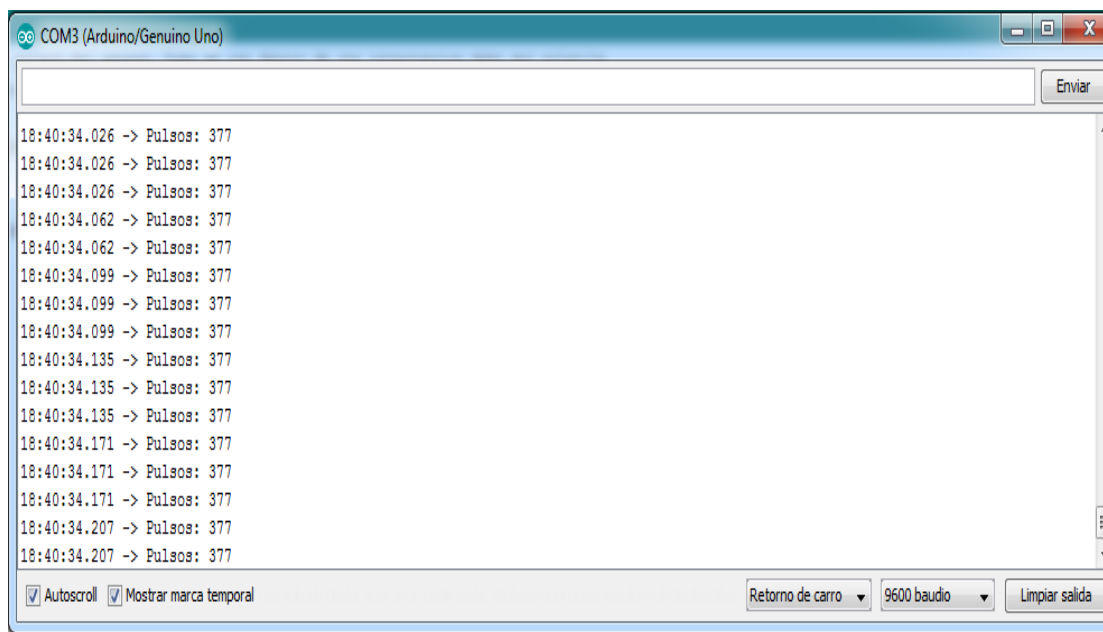
**Figura 4. 19.- Prueba 1 de calibración – Volumen un litro.
(Captura de pantalla)**

En la prueba 2 se obtuvo 377 pulsos, como muestra a figura 4.20.



**Figura 4. 20.- Prueba 2 de calibración – Volumen un litro.
(Captura de pantalla)**

En la prueba 3 se obtuvo 377 pulsos, como muestra a figura 4.21.



**Figura 4. 21.- Prueba 3 de calibración – Volumen un litro.
(Captura de pantalla)**

En la tabla 4.1 nuestra un resumen de los datos que se obtuvieron, se aplicó la fórmula para hallar el factor de conversión (K) y finalmente nos quedamos con el promedio

- Prueba ½ Lt: $189 \times 2 = 378$ pulsos x Litro ; $K=6.30$
- Prueba 1Lt: 378 Pulsos por Litro; $K=6.30$

Entonces podemos inferir que por cada litro de agua que pasa por el sensor obtenemos aproximadamente **378 pulsos** con un **factor de conversión 6.30** .

Tabla 4. 1.- Resumen de datos de la prueba de calibración.

Litros	N° Pulsos	K
½ Lt	188	6.27
½ Lt	190	6.33
½ Lt	189	6.30
Promedio ½ Lt	189	6.30
1 Lt	380	6.33
1 Lt	377	6.29
1 Lt	377	6.29
Promedio 1 Lt	378	6.30

(Elaboración de los autores)

4.3.2. Implementación de hardware.

En la implementación de hardware se siguieron los siguientes pasos:

- Se insertó un SIM en el Shield, como lo muestra la figura 4.22.

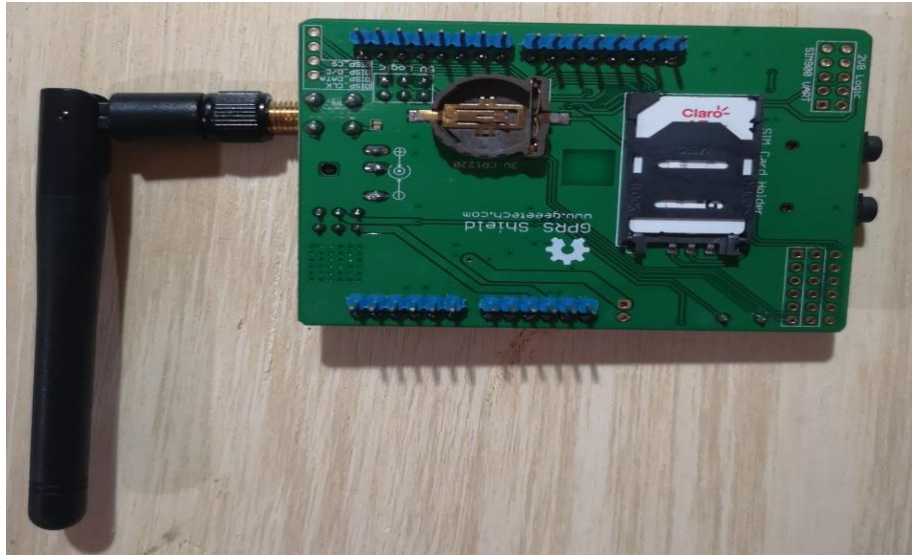


Figura 4. 22.- Inserción del SIM Card al Shield GSM/GPRS SIM900.
(Captura fotográfica de los autores)

- Se procedió a conectar los pines del Shield al Arduino Uno, debe conectarse como muestra la figura 4.23.



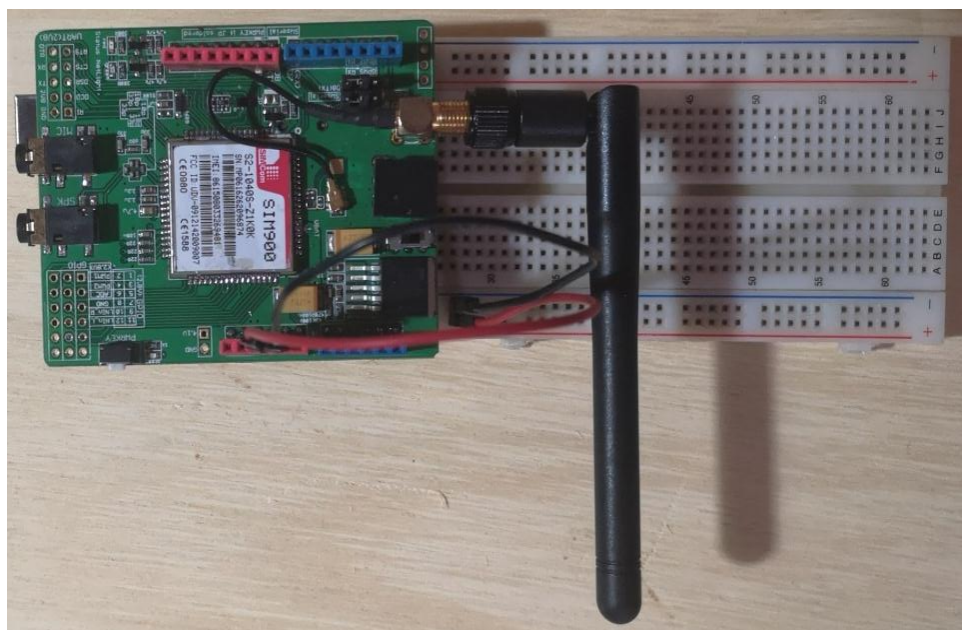
Figura 4. 23.- Ubicación para Inserción del Shield GSM/GPRS SIM900 al Arduino.
(Captura fotográfica de los autores)

- Quedando conectado como muestra la figura 4.24.



**Figura 4. 24.- Inserción del Shield GSM/GPRS SIM900 al Arduino.
(Captura fotográfica de los autores)**

- Luego lo montamos en el protoboard, y conectamos un cable rojo del protoboard a 5V del SIM 900, y un cable negro al GND del Shield, como lo muestra la figura 4.25.



**Figura 4. 25.- Montado y conexión de Shield SIM900/Arduino al Protoboard.
(Captura fotográfica de los autores)**

- En el siguiente paso se conectó el modulo relé al protoboard, donde el pin de la izquierda se conecta al GND (cable negro), pin del medio al 5V (cable rojo) del protoboard y pin de la derecha de la señal (cable amarillo), va al pin 12 del Shield, como muestra la figura 4.26.

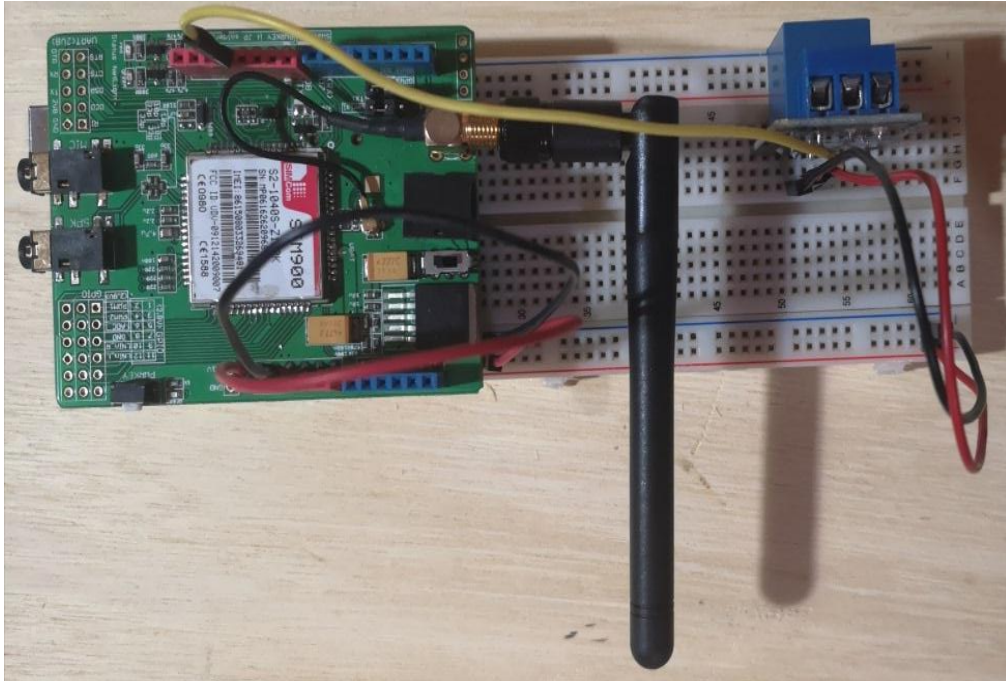
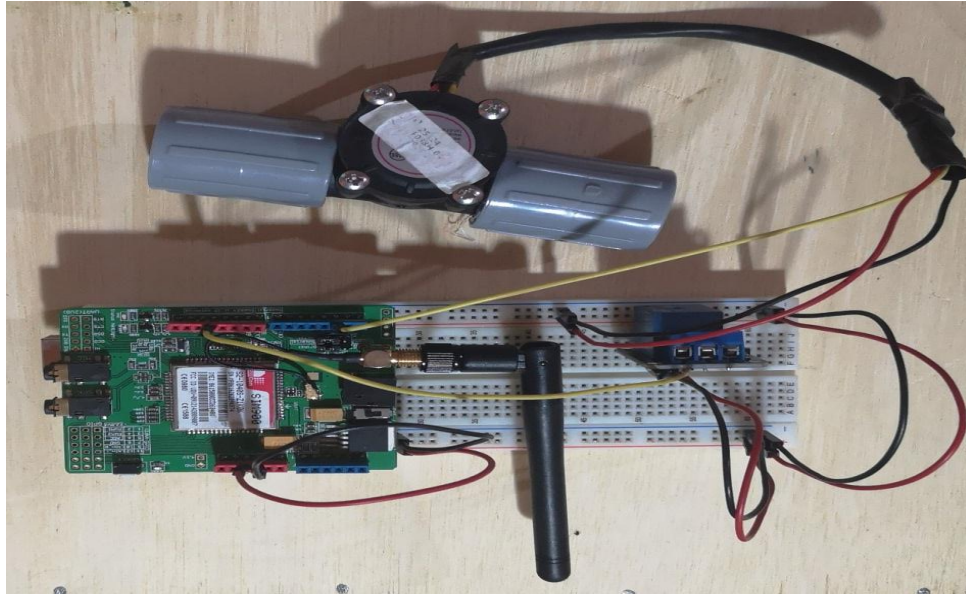


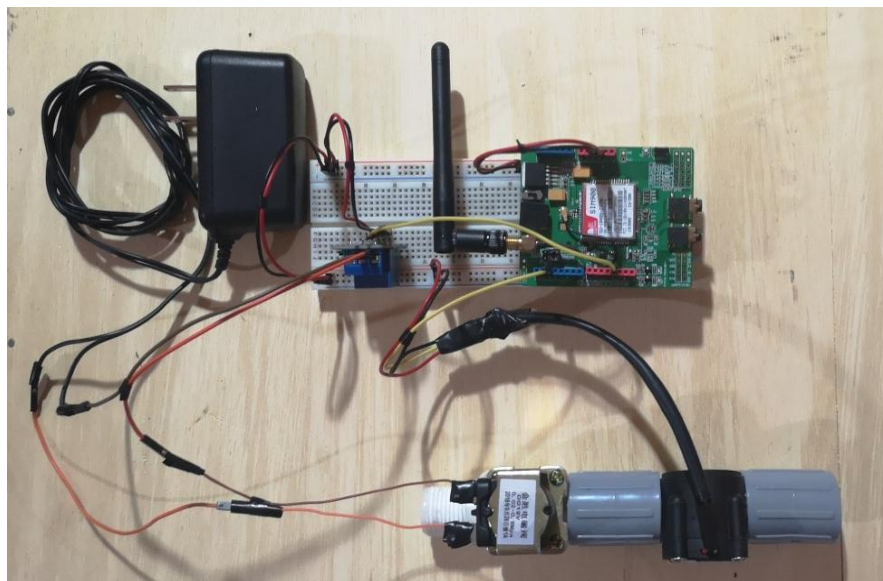
Figura 4. 26.- Conexión del Relé.
(Captura fotográfica de los autores)

- Se procedió a conectar el sensor de caudal de agua, donde el cable rojo y negro del sensor van conectado al 5V y GND del protoboard respectivamente, y el cable amarillo del sensor al pin 2 del Shield; como se muestra en la figura 4.27.



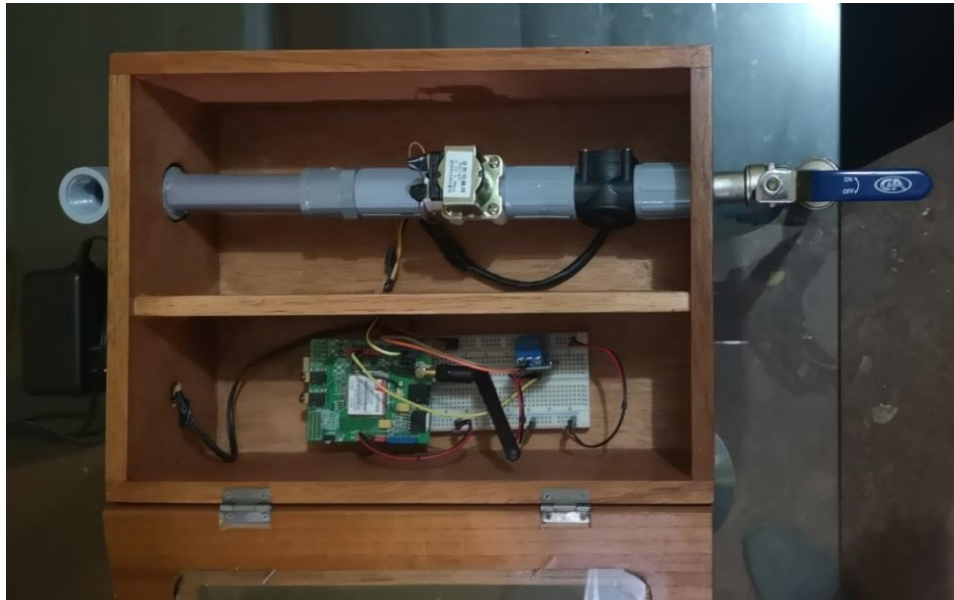
**Figura 4. 27.- Conexión del sensor de caudal de agua.
(Captura fotográfica de los autores)**

- En el siguiente paso se hizo la conexión del cargador con la electroválvula a la entradas del relé, donde entrada de la derecha del relé (cable anaranjado) va conectado a un pata de la electroválvula (cable marrón), la otra pata de la electroválvula se conectada a un polo del cargador (positivo), y el otro polo del cargador (negativo) a la entrada del centro del relé, quedando como muestra la figura 4.28.



**Figura 4. 28.- Conexión de la electroválvula y cargador hacia el relé.
(Captura fotográfica de los autores)**

- Por último, se montó el circuito a la caja y se hicieron las conexiones de los accesorios de gasfitería, quedando como muestra la figura 4.29 y 4.30.



**Figura 4. 29.- Montado de circuito a la caja y conexión de accesorios de gasfitería.
(Captura fotográfica de los autores)**



**Figura 4. 30.- Hardware terminado.
(Captura fotográfica de los autores)**

4.3.3. Implementación de software.

Para la implementación de Software se trabajó en la plataforma Arduino versión 1.8.8 y a continuación se explicará paso a paso su implementación.

- En primer lugar se declaró todas la variables, que se utilizarán, como lo muestra la figura 4.31 y que detallaremos a continuación:
 - ✓ SoftwareSerial SIM900(6,7); es para configurar e indicar que el SIM900 va a trabajar con el pin 6 y 7 del Arduino.
 - ✓ Respuesta; variable que indica la respuesta del mensaje de texto.
 - ✓ Aux_str[50]; almacena temporalmente el mensaje de texto.
 - ✓ Sms[]="LLAVE ABIERTA \x1A \r\n"; almacena el mensaje a responder "LLAVE ABIERTA."
 - ✓ Smc[]="LLAVE CERRADA \x1A \r\n"; almacena el mensaje a responder "LLAVE CERRADA."
 - ✓ Mcubicos[]="m3"; almacena la unidad de medida del agua.
 - ✓ Smsprint[]="\x1A \r\n"; almacena comando para imprimir el mensaje de texto.
 - ✓ Convertirchar[50]; almacena el valor del volumen convertido de float a char.
 - ✓ Concatenar[80]; almacena el texto concatenado a responder con el volumen de agua consumido.
 - ✓ Mensaje_recibido=0; almacena 0 o 1 para verificar si el mensaje fue recibido.

- ✓ Relay=12; almacena el número del pin al que va conectado el relé.
- ✓ Pulsos; almacena el número de pulsos que da el sensor de caudal de agua.
- ✓ Litrosporhora; almacena los litros por hora.
- ✓ SensorDeFlujo=2; almacena el número de pin al que va conectado el sensor de flujo de agua.
- ✓ TiempoAnterior; almacena el tiempo que se tiene que calcular para los litros de agua.
- ✓ PulsosAcumulados; almacena los sumatoria de los pulsos que se van obteniendo del sensor de flujo de agua.
- ✓ Metroscubicos; almacena los metros cúbicos que pasan por el sensor de flujo de agua en float.

```

#include <SoftwareSerial.h>
SoftwareSerial SIM900(7, 8); // Configura el puerto serial para el SIM900.
int respuesta;
char aux_str[50];
//Contenido del sms que enviamos. \x1A corresponde al caracter de finalizacion
char smsa[] = "LLAVE ABIERTA \x1A \r\n";
char smsc[] = "LLAVE CERRADA \x1A \r\n";
char mcubicos[]=" m3";
char smsprint[]=" \x1A \r\n";
char convertirchar[50];
char concatenar[80];
char mensaje_recibido=0;
int relay=12;
int pos_memoria= 0;
volatile int pulsos; // Cantidad de pulsos del sensor. Como se usa dentro de una interrupcion debe ser volatile
unsigned int litrosPorHora; // Calculated litres/hour
unsigned char sensorDeFlujo = 2; // Pin al que esta conectado el sensor
unsigned long tiempoAnterior; // Para calcular el tiempo
unsigned long pulsosAcumulados; // Pulsos acumulados
float metrosCubicos; // Metros cubicos acumulados

```

**Figura 4. 31.- Inicialización de variables.
(Captura de pantalla)**

- Luego se procedió a crear la función **flujo()**, que calcula el número de pulsos y la función **calcular()**, que permite calcular los litros por hora que luego son transformados a volumen en metros cúbicos, para ello se utilizó los datos ya calculados en la calibración del sensor de caudal de agua, que figuran en la tabla 4.1:

- ✓ N° de pulsos por litro=378
- ✓ El factor conversión =6.30

Así lo demuestra la figura 4.32.

```

void flujo () // Funcion de interrupcion
{
    pulsos++; // Simplemente sumar el numero de pulsos
}

void calcular()
{
    if( millis() - tiempoAnterior > 1000)
    {
        tiempoAnterior = millis(); // Updates cloopTime
        pulsosAcumulados += pulsos;
        litrosPorHora = (pulsos * 60 / 6.30); // (Pulso de frecuencia x 60 min) / 7.5Q = flow rate in L/hour
        pulsos = 0; // Reset Counter
        metrosCubicos = ((pulsosAcumulados*1.00/378)/1000); //Cada 378 pulsos = 1 litro
    }
}

```

**Figura 4. 32.- Calcular N° de pulsos y Volumen.
(Captura de pantalla)**

- Luego en la función void setup() se procedió a inicializar:
 - ✓ **SIM900.begin(19200):** Velocidad del puerto serie para el SIM900 a 19200
 - ✓ **Serial.begin(19200):** Velocidad del puerto serie del Ardino a 19200.
 - ✓ **pinMode():** Entrada para el sensor de flujo de agua(pin2) y de salida para relé (pin12).
 - ✓ **digitalWrite():** Activamos el relé para que la electrovalvula encienda, y permita el paso de agua.
 - ✓ **attachInterrupt():** Interrupción para los pulsos.
 - ✓ **millis():** tiempo en milisegundos.
 - ✓ **power_on():** funcion para iniciar el SIM900.

- ✓ **Iniciar():** Función para activar el envío o recepción de mensajes de texto.

Como lo muestra la figura 4.33.

```
void setup()
{
  SIM900.begin(19200); //Configura velocidad del puerto serie para el SIM900
  Serial.begin(19200); //Configura velocidad del puerto serie del Arduino
  pinMode(sensorDeFlujo, INPUT);
  pinMode(relay, OUTPUT);
  digitalWrite(relay, HIGH);
  attachInterrupt(0, flujo, RISING); // Interrupción para los pulsos
  interrupts(); // Habilitar interrupciones
  tiempoAnterior = millis();
  Serial.println("Iniciando...");
  power_on();
  iniciar();
}
```

**Figura 4. 33.- Función para inicialización de variables.
(Captura de pantalla)**

- Luego de inicializar las variables en void setup(), se procedió a hacer uso de las variables inicializadas en la función void loop() , en dónde.
 - ✓ Se llama la función calcular(), el cual calcula el volumen en metros cúbicos.
 - ✓ Se verifica si se recibe un mensaje de texto.
 - ✓ Si se recibe un mensaje texto responde a 3 opciones:
 - Si el mensaje de texto es 'A', enciende el relé y por ende se enciende la electroválvula,

respondiendo con un mensaje de texto “LLAVE ABIERTA” con la función mensaje_sms(),

- Si el mensaje de texto es ‘E’, apaga el relé y por ende se apaga la electroválvula, respondiendo con un mensaje de texto “LLAVE CERRADA” con la función mensaje_sms().
- Si el mensaje de texto es ‘R’, responde con un mensaje de texto con el consumo de agua en metros cúbicos, con la función mensaje_sms().

Así lo demuestra la figura 4.34.

```
void loop()
{
  calcular();
  if (SIM900.available()) {
    mensaje_recibido=SIM900.read();
    switch (mensaje_recibido)
    {
      case 'A':// Si el mensaje es 'A' procede a abrir electroválvula
        digitalWrite(relay, HIGH);
        mensaje_sms(smsa);// Envia mensaje "LLAVE ABIERTA"
        break;
      case 'E'://Si el mensaje es 'E' procede a cerrar electroválvula
        digitalWrite(relay, LOW);
        mensaje_sms(smsc);//Envia mensaje "LLAVE CERRADA"
        break;
      case 'R'://Si el mensaje es 'R' envia el consumo de agua
        dtostrf(metroscubicos, 6, 3, convertirchar);//Conversión del volumen a char
        sprintf(concatenar,"%s %s %s",convertirchar,mcubicos,smsprint);//concatenación
        mensaje_sms(concatenar);//envia mensaje con el volumen consumido
        break;
    }
  }
}
```

**Figura 4. 34.- Función loop - Envío de mensajes de texto.
(Captura de pantalla)**

- La función enviarAt(), recibe los comandos AT y verifica si son correctos, como lo muestra la figura 4.35.

```

int enviarAT(char* ATcommand, char* resp_correcta, unsigned int tiempo)
{
    int x = 0;
    bool correcto = 0;
    char respuesta[100];
    unsigned long anterior;
    memset(respuesta, '\0', 100); // Inicializa el string
    delay(100);
    while ( SIM900.available() > 0) SIM900.read(); // Limpia el buffer de entrada
    SIM900.println(ATcommand); // Envia el comando AT
    x = 0;
    anterior = millis(); // Espera una respuesta
    do { // si hay datos el buffer de entrada del UART lee y comprueba la respuesta
        if (SIM900.available() != 0)
        { // Comprueba que no haya desbordamiento en la capacidad del buffer
            if (x < 99) {
                respuesta[x] = SIM900.read();
                x++;
            }
            else Serial.println("Desbordamiento!");
            // Comprueba si la respuesta del modulo es la 1
            if (strstr(respuesta, resp_correcta) != NULL)
            {
                correcto = 1;
            }
        }
    } // Espera hasta tener una respuesta
    while ((correcto == 0) && ((millis() - anterior) < tiempo));
    Serial.println(respuesta);
    return correcto;
}

```

**Figura 4. 35.- Verificación de comandos AT.
(Captura de pantalla)**

- Las función power_on() y power_off, enciende y apaga el SIM900, respectivamente, como lo muestra la figura 4.36.

```

void power_on()
{
    int respuesta = 0;
    // Comprueba que el modulo SIM900 esta arrancado
    if (enviarAT("AT", "OK", 2000) == 0) //comprueba la conexion a la red
    {
        Serial.println("Encendiendo el GPRS...");
        pinMode(9, OUTPUT);
        digitalWrite(9, HIGH);
        delay(1000);
        digitalWrite(9, LOW);
        delay(1000);
        // Espera la respuesta del modulo SIM900
        while (respuesta == 0) {
            // Envia un comando AT cada 2 segundos y espera la respuesta
            respuesta = enviarAT("AT", "OK", 2000);
            SIM900.println(respuesta);
        }
    }
}

void power_off()
{
    digitalWrite(9, HIGH);
    delay(1000);
    digitalWrite(9, LOW);
    delay(1000);
}

```

**Figura 4. 36.- Encendido y Apagado del SIM900.
(Captura de pantalla)**

- La función reiniciar(), reinicia el SIM900 y la función iniciar(), envía comandos AT, activando el envío y recepción de mensajes de texto, como lo muestra la figura 4.37.

```

void reiniciar()
{
    Serial.println("Reiniciando...");
    power_off();
    delay (5000);
    power_on();
}
void iniciar()
{
    enviarAT("AT+CPIN=\"1867\"", "OK", 1000);
    Serial.println("Conectando a la red...");
    delay (5000);

    //espera hasta estar conectado a la red movil
    while ( enviarAT("AT+CREG?", "+CREG: 0,1", 1000) == 0 )
    { }
    Serial.println("Conectado a la red.");
    enviarAT("AT+CLIP=1\r", "OK", 1000); // Activamos la identificacion de llamadas
    enviarAT("AT+CMGF=1\r", "OK", 1000); //Configura el modo texto para enviar o recibir mensajes
    enviarAT("AT+CNMI=2,2,0,0,0\r", "OK", 1000); //Configuramos el modulo para que nos muestre los SMS recibidos por comunicacion serie
    Serial.println("Preparado.");
}

```

**Figura 4. 37.- Envio de comandos AT.
(Captura de pantalla)**

- La función `mensaje_sms()`, envía mensajes de texto a un número de celular al cual se le indica, como lo muestra la figura 4.38.


```

void mensaje_sms(char* sms)
{
    if (enviarAT("AT+CREG?", "+CREG: 0,1", 1000) == 1) //comprueba la conexion a la red
    {
        Serial.println("Enviando SMS...");
        enviarAT("AT+CMGF=1\r", "OK", 1000); //Comando AT para mandar un SMS
        sprintf(aux_str, "AT+CMGS=\"962644717\"", strlen(sms)); //Numero al que vamos a enviar el mensaje
        //Texto del mensaje
        if (enviarAT(aux_str, ">", 100) == 1)
        {
            enviarAT(sms, "OK", 10000);
        }
        Serial.println("SMS enviado");
    }
    else
    {
        reiniciar();
        iniciar();
    }
}

```

**Figura 4. 38.- Envío de mensajes de texto.
(Captura de pantalla)**

4.3.4. Integración de hardware y software.

En esta parte se integró el hardware y software, siguiendo los siguientes pasos:

- En primer lugar se conectaron los cargadores de 5 voltios (Arduino) y 12 voltios (electroválvula) a la corriente, como también el cable USB de la laptop a la placa Arduino, para que muestre en pantalla de la laptop el correcto funcionamiento del prototipo, como se muestra la figura 4.39.



**Figura 4. 39.- Conexión para la integración del prototipo.
(Captura fotográfica de los autores)**

- Una vez que se encontraba todo conectado se procedió a subir el programa al Arduino, en la figura muestra un mensaje en la parte inferior izquierda un mensaje “Subido”, que indica que el programa ha subido correctamente, como se muestra la figura 4.40.

```
ConsumoAguaPotable $
#include <SoftwareSerial.h>
SoftwareSerial SIM900(7, 8); // Configura el puerto serial para el SIM900.
int respuesta;
char aux_str[50];
//Contenido del sms que enviamos. \x1A corresponde al caracter de finalizacion
char smsa[] = "LLAVE ABIERTA \x1A \r\n";
char smsc[] = "LLAVE CERRADA \x1A \r\n";
char mcubicos[]=" m3";
char smsprint[]=" \x1A \r\n";
char convertirchar[50];
char concatenar[80];
char mensaje_recibido=0;
int relay=12;
int pos_memoria= 0;
volatile int pulsos; // Cantidad de pulsos del sensor. Como se usa dentro de una interrupcion debe ser volatile
unsigned int litrosPorHora; // Calculated litres/hour
unsigned char sensorDeFlujo = 2; // Pin al que esta conectado el sensor
unsigned long tiempoAnterior; // Para calcular el tiempo
unsigned long pulsosAcumulados; // Pulsos acumulados
float metrosCubicos; // Metros cubicos acumulados

void flujo () // Funcion de interrupcion
{
    pulsos++; // Simplemente sumar el numero de pulsos
}

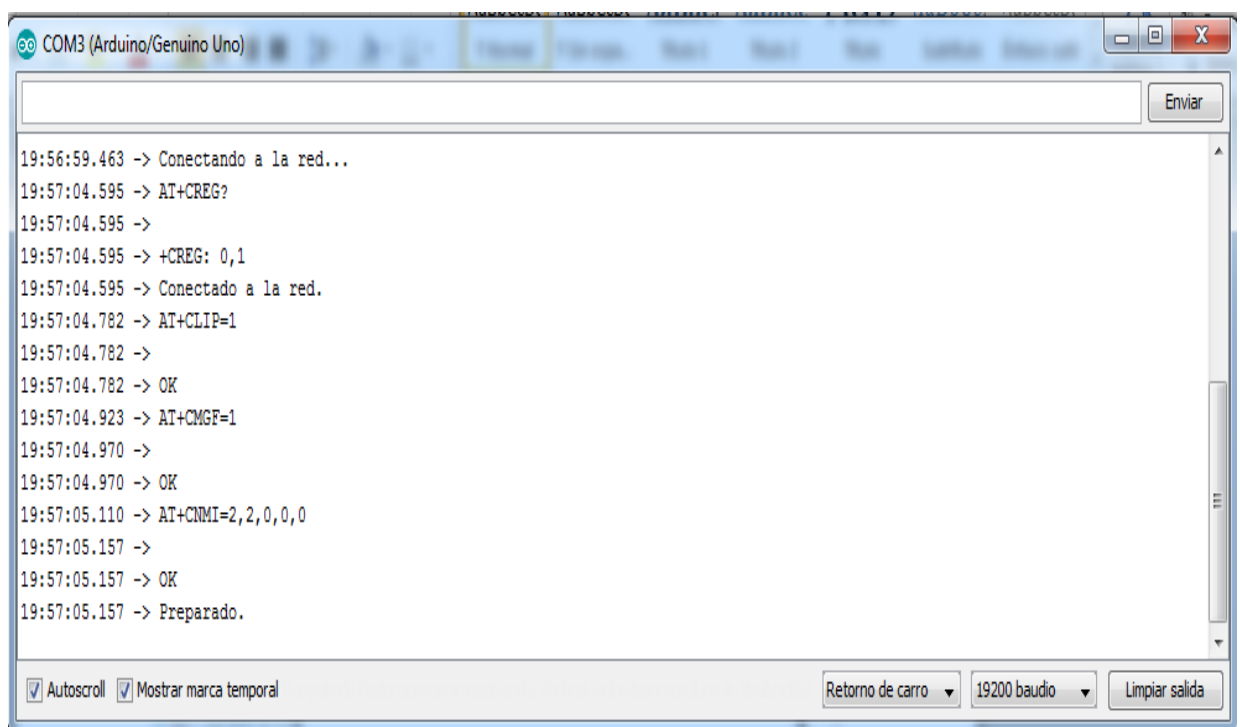
void calcular()
{
    if( millis() - tiempoAnterior > 1000)
```

**Figura 4. 40.- Subida del programa a la placa de Arduino.
(Captura de pantalla)**

- Finalmente se abrió el monitor serie para ver corroborar que la integración fue exitosa, en la figura 4.41 muestra que está iniciando el SIM900 y en la figura 4.42 muestra que el SIM900 está preparado para recibir mensajes de texto.



**Figura 4. 41.- Inicio del SIM900.
(Captura de pantalla)**



**Figura 4. 42.- SIM900 Preparado.
(Captura de pantalla)**

4.3.5. Funcionamiento y resultados del prototipo.

4.3.5.1. Funcionamiento.

Una vez que se tuvo el Prototipo listo se puso en funcionamiento para el monitorear y controlar el consumo de agua potable, se recuerda que el SIM900 reconoce tres mensajes de texto:

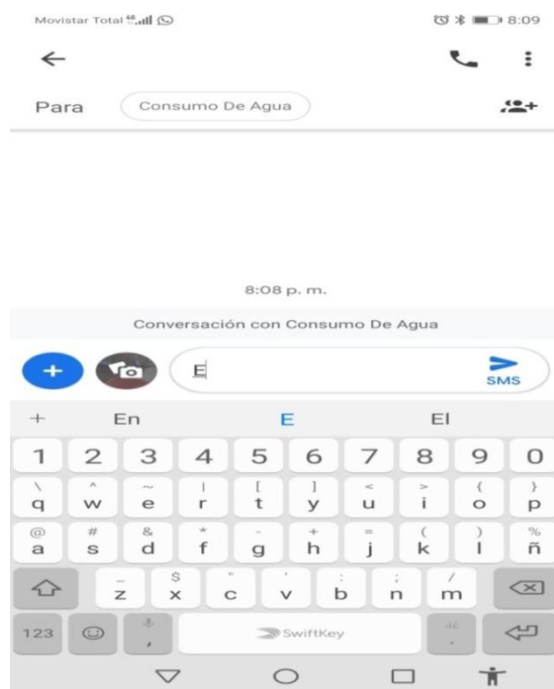
- ‘A’: Para abrir la electroválvula y permitir el fluido de agua potable.
- ‘E’: Para cerrar la electroválvula y cortar el fluido de agua potable.
- ‘R’: Para conocer el consumo de agua potable.

En la figura 4.43 muestra el prototipo en funcionamiento, en el cual se observa que está llenando un balde a agua, es decir, que al iniciar el Arduino por primera vez, automáticamente se abre la electroválvula permitiendo el fluido de agua potable.



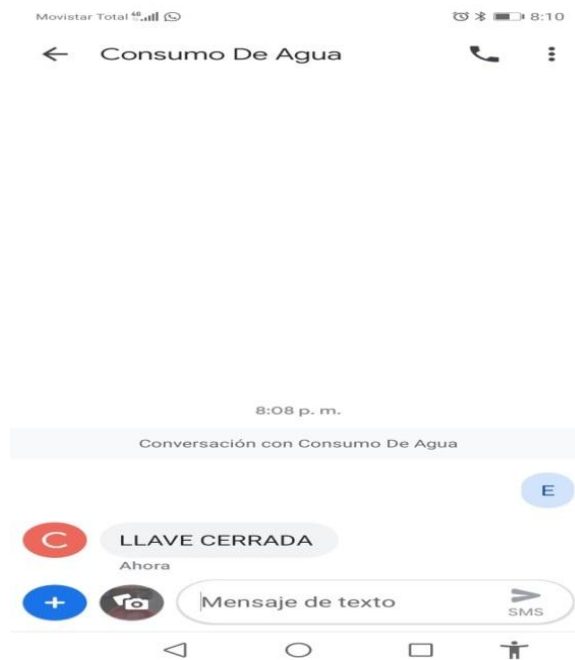
**Figura 4. 43.- Muestra de fluido de agua
(Captura fotográfica de los autores.)**

Se envió un mensaje de texto con la letra 'E', desde el teléfono celular que se configuró, para cerrar la electroválvula y cortar el fluido de agua como muestra la figura 4.44.



**Figura 4. 44.- Envío de mensaje con la letra 'E'.
(Captura de pantalla de teléfono celular)**

Se recepcionó un mensaje de texto que indicó que la llave se ha cerrado como muestra la figura 4.45.



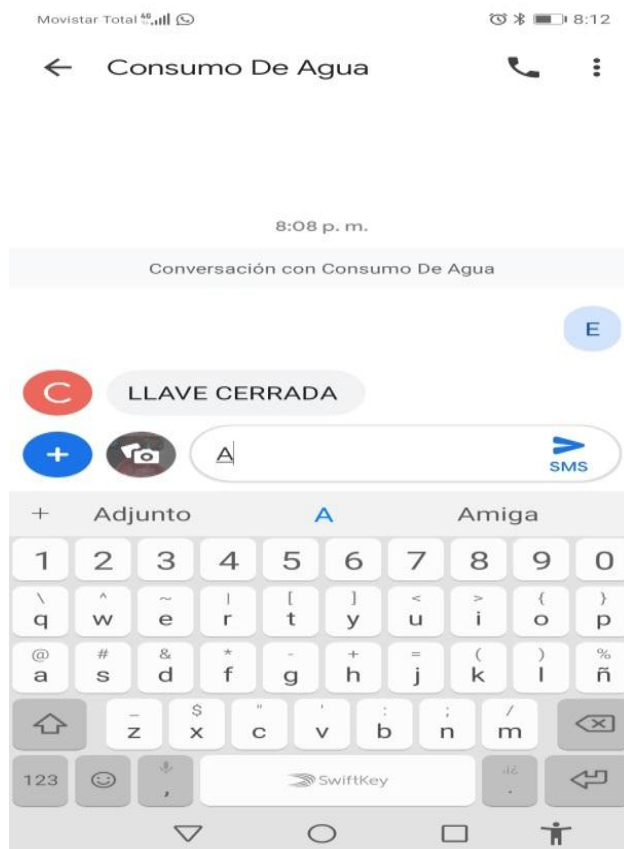
**Figura 4. 45.- Recepción de mensaje 'LLAVE CERRADA'.
(Captura de pantalla de teléfono celular)**

En la figura 4.46 muestra que se cortó el fluido de agua, como se puede observar, la electroválvula se ha cerrado.



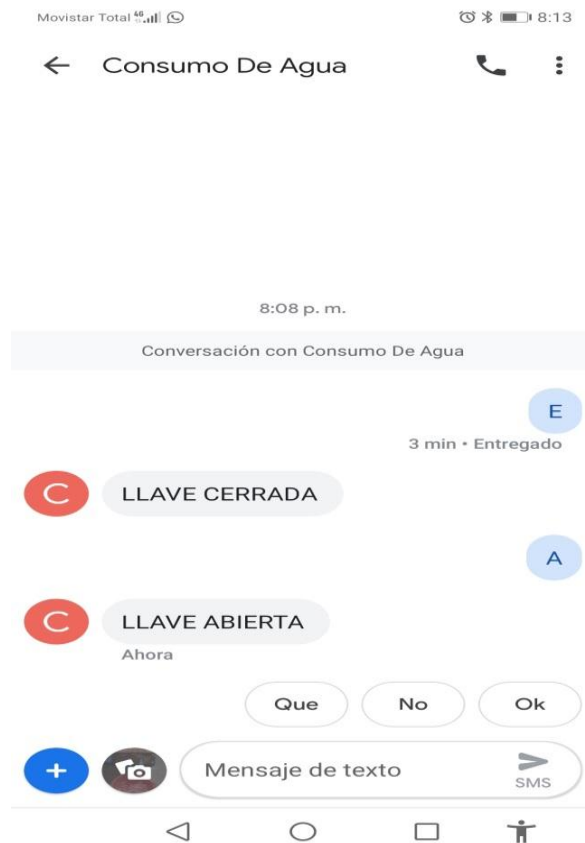
**Figura 4. 46.- Corte de fluido de agua.
(Captura fotográfica de los autores.)**

Luego para abrir la electroválvula y permitir el fluido de agua potable se envió un mensaje de texto con la letra 'A', como muestra la figura 4.47.



**Figura 4. 47.- Envío de mensaje con la letra 'A'.
(Captura de pantalla de teléfono celular)**

Se recepcionó un mensaje de texto que indica que la llave se ha abierto como muestra la figura 4.48.



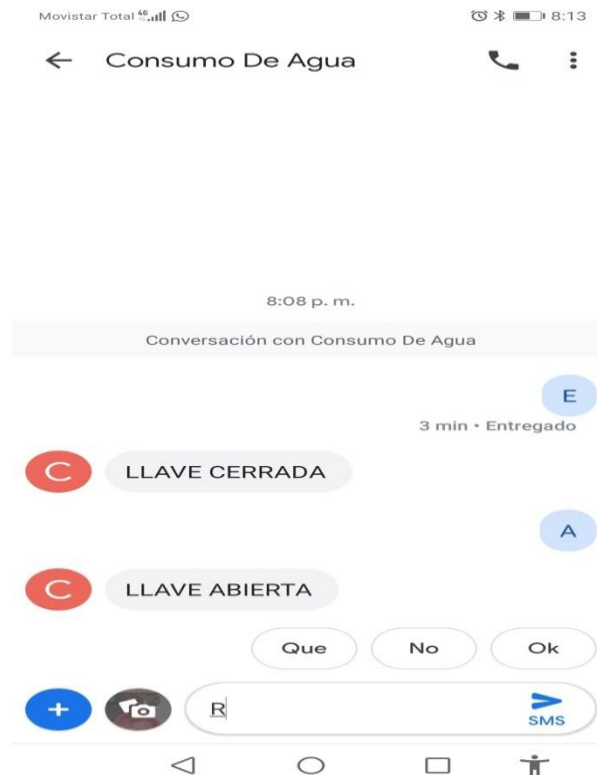
**Figura 4. 48.- Recepción de mensaje 'LLAVE ABIERTA'.
(Captura de pantalla de teléfono celular)**

En la figura 4.49 muestra que volvió el fluido de agua, como se puede observar, la electroválvula se ha abierto.



**Figura 4. 49.- Regreso de fluido de agua.
(Captura fotográfica de los autores.)**

Luego conocer el consumo de agua potable, se envió un mensaje de texto con la letra '**R**', como muestra la figura 4.50.



**Figura 4. 50.- Envío de mensaje con la letra ‘E’.
(Captura de pantalla de teléfono celular)**

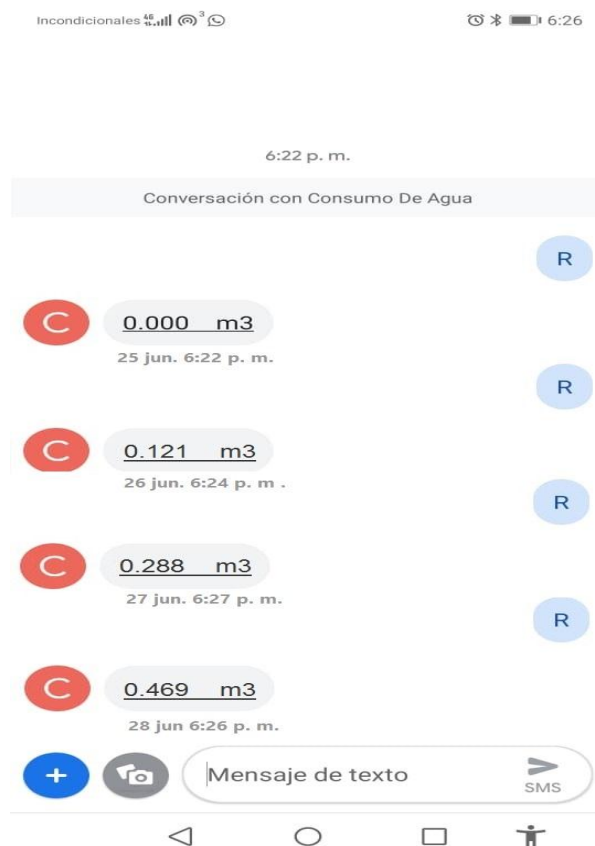
Finalmente, se recepciona un mensaje indicando el consumo de agua potable, en ese momento el consumo es de 0.032 m³, como lo muestra la figura 4.51.



**Figura 4. 51.- Recepción de mensaje con el consumo de agua.
(Captura de pantalla de teléfono celular)**

4.3.5.2. Resultados.

Se realizaron pruebas del consumo de agua potable durante tres días consecutivos, pero sin controlar la apertura y cierre de la electroválvula desde el 25 hasta el 28 de junio de 2019, empezando la lectura del agua desde las 06:22 pm, y consultando el consumo de agua entre 6:22 pm y 6:27 pm durante los días mencionados, con el objetivo de saber solo el consumo de agua, se le indicó a las personas que viven en la vivienda que utilicen al agua como normalmente lo hacen, en la figura 4.52 muestran los datos de consumo de agua por los días antes mencionado.



**Figura 4. 52.- Consumo de agua potable, sin controlar.
(Captura de pantalla de teléfono celular)**

Se realizaron pruebas del consumo de agua potable durante tres días consecutivos, pero sin controlar la apertura y cierre de la electroválvula, en resumen, se obtuvieron los datos de la tabla 4.2.

Tabla 4. 2.-Resumen de datos de consumo de agua sin controlar.

Día	1	2	3
Consumo de agua acumulativa (m ³)	0.121	0.288	0.469
Consumo por día (m ³)	0.121	0.167	0.181

(Elaboración de los autores)

En el grafico muestra el consumo de agua por día

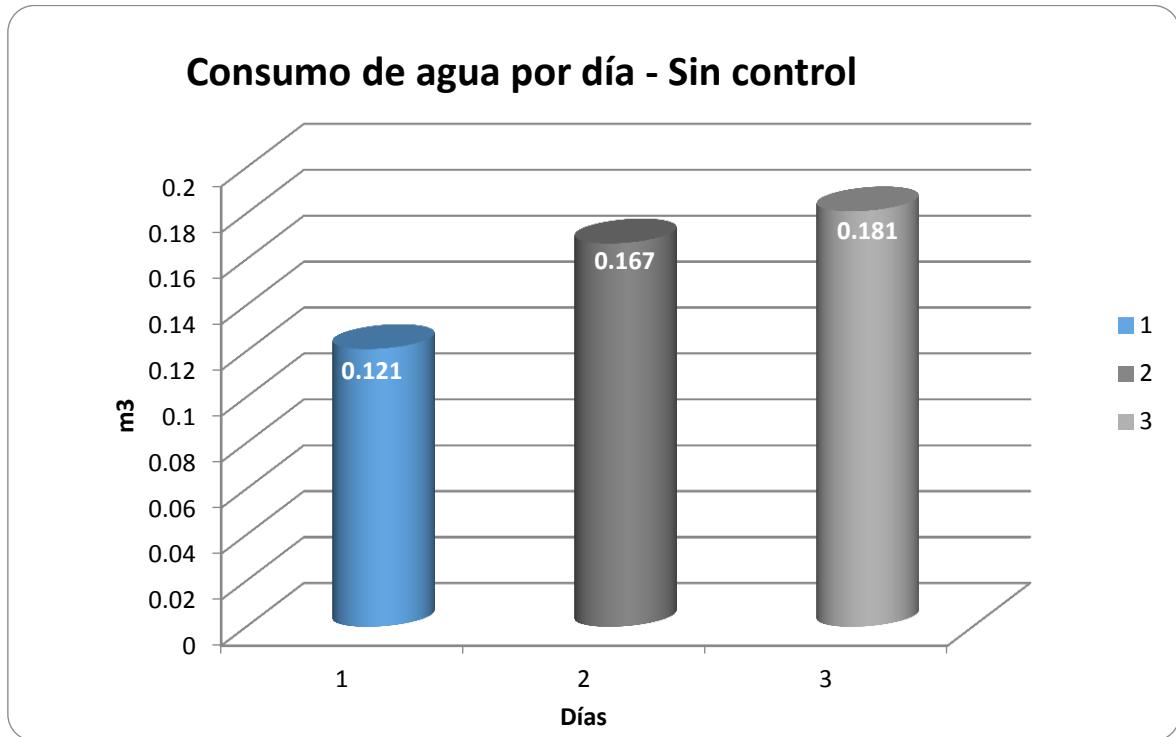


Gráfico 4. 1.- Consumo de agua por día sin controlar
Fuente: Elaboración en Microsoft Excel

Luego se realizaron pruebas de consumo de agua por tres días consecutivos más, desde el 28 de junio al 1 de julio de 2019, consultando el consumo de agua entre 6:30 pm a 6:36 pm durante los días mencionados, esta vez realizando el control de apertura y cierre de agua mediante un mensaje de texto, como lo muestra la figura 4.53 y figura 4.54.



Figura 4. 53.- Consumo de agua potable controlado parte1.
(Captura de pantalla de teléfono celular)

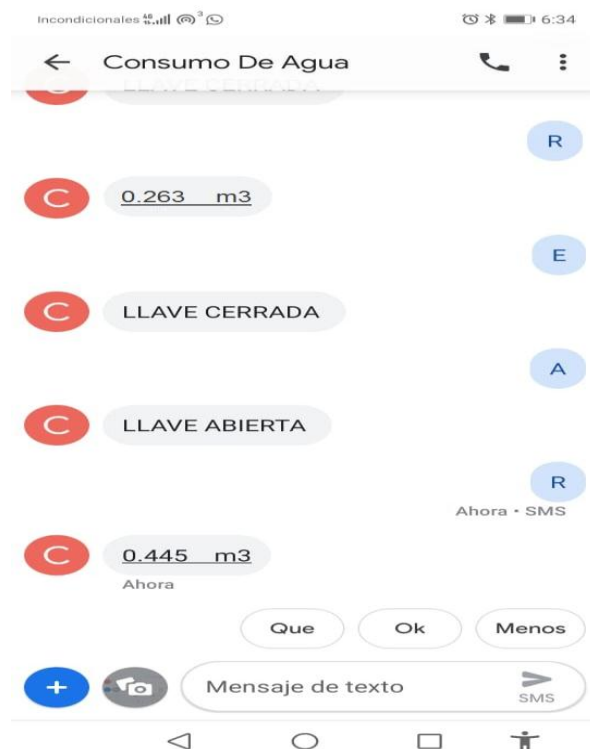


Figura 4. 54.- Consumo de agua potable controlado parte2.
(Captura de pantalla de teléfono celular)

En la tabla 4.3 se tiene un resumen de los datos obtenidos en el consumo de agua controlado.

Tabla 4. 3.-Resumen de datos de consumo de agua controlado.

Día	1	2	3
Consumo de agua acumulativo (m ³)	0.117	0.263	0.445
Consumo por día (m ³)	0.117	0.146	0.182

(Elaboración de los autores)

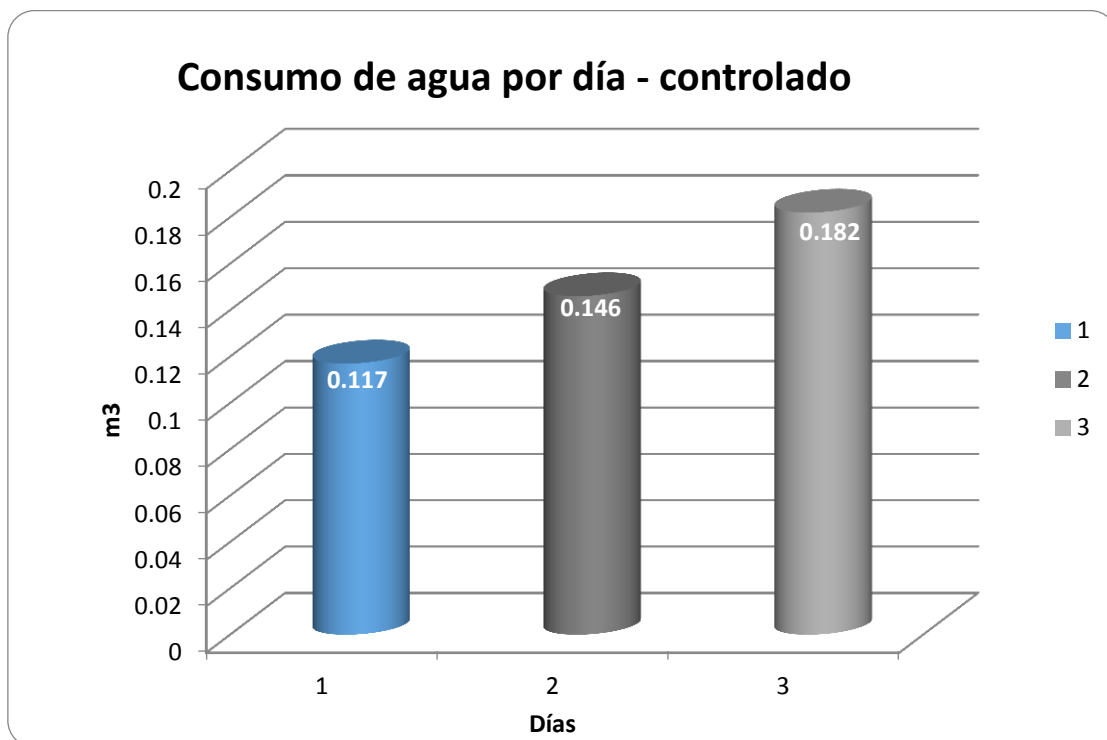


Gráfico 4. 2.- Consumo de agua por día, controlado

Fuente: Elaboración en Microsoft Excel

Luego en la tabla 4.4 muestra un comparativo del consumo de agua acumulativa controlada y sin controlar.

Tabla 4. 4.-Cuadro comparativo de consumo de agua acumulativa.

Día	1	2	3
Sin Control	0.121	0.288	0.469
Con Control	0.117	0.263	0.445

(Elaboración de los autores)

El grafico 4.3 del comparativo de consumo de agua acumulativo muestra que se redujo el consumo de agua de 0.469 m³ a 0.445 m³, obteniendo un ahorro de agua de 0.024 m³, que es un 5% del agua consumida solo en 3 días.

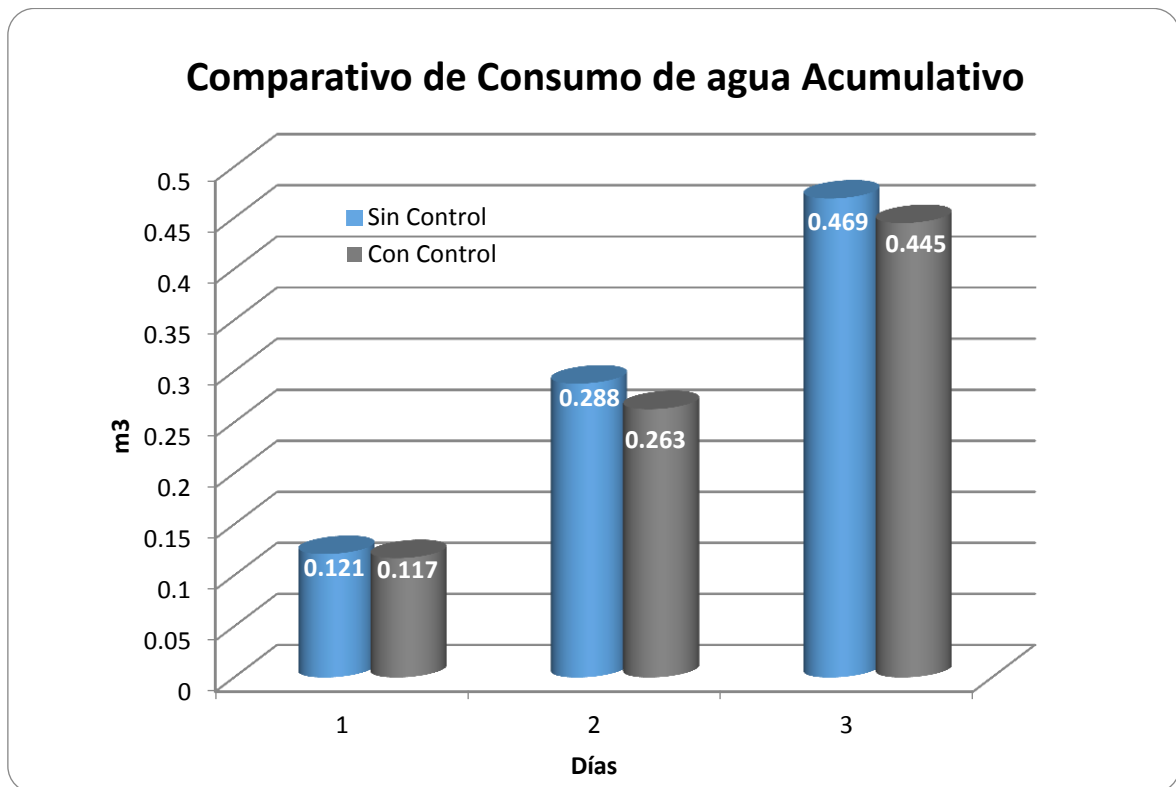


Gráfico 4. 3.- Comparativo de agua potable acumulativo.
Fuente: Elaboración en Microsoft Excel

CONCLUSIONES

- Analizando los datos obtenidos, se concluye que se evita el desperdicio de agua llevando un control de apertura y cierre de la electroválvula, ahorrando un 5% de agua potable.
- La aplicación de la tecnología GSM permitió conocer el consumo de agua potable de manera inmediata por mensaje de texto ya que esta tecnología lo permite, como también poder controlar el consumo.
- La implementación del prototipo basado en una línea de investigación, del Internet de las Cosas (IoT), incidió positivamente en la automatización del control de consumo, cada hardware utilizado como son el sensor y la válvula solenoide fueron los principales actores.
- Con el teléfono móvil se pudo realizar el adecuado monitoreo y control del consumo de agua potable, ya que permitió consultar y controlar desde cualquier distancia y en cualquier momento.

RECOMENDACIONES

- Para futuros proyectos, se recomienda trabajar con tarjeta que incorpore las funciones de Arduino, del Wifi y la corriente (baterías).
- Incorporar plataformas de protección para evitar la humedad, para proteger a los circuitos y no exista interferencia.
- Diseñar un aplicativo que implemente un sistema de alerta en caso de fugas de agua potable, periodo de reportes, riesgos de consumo, etc.
- Socializar el proyecto con los ciudadanos, haciendo énfasis en las ventajas que tiene el proyecto y lo económico que es para adquirirlo.
- El proyecto de investigación esta aplicado para hogares, pero para futuras aplicaciones se puede realizar a instituciones, colegios, universidades o empresas.

REFERENCIAS BIBLIOGRÁFICAS

Arduino Uno R3. Disponible en <https://www.infootec.net/arduino/> [accesado el 03 de mayo de 2019.]

Artero, Óscar Torrente. (2013). ARDUINO: Curso práctico de formación. México.

Autoridad Nacional de Agua – ANA: El agua en cifras. Disponible en <http://ana.gob.pe/contenido/el-agua-en-cifras> [accesado el 03 de mayo de 2019]

Banzi, Massimo. (2013). Introducción a Arduino.

Castro, A. (2000). Sistema de Control de Temperatura a través de Arduino y la Tecnología GPRS/GSM. Tesis. Licenciado en Ingeniería y Sistema de Telecomunicación. Universidad Politécnica de Madrid. Colección de tesis digitales UPM.

Cembranos, F. (2013). UF1796 - Planificación de la gestión y organización de los procesos de montaje de sistemas de automatización industrial.

Evans, D. (2011). Internet of Things - La próxima evolución de Internet lo está cambiando todo.

Evans, Brian W. (2013). Arduino Programming Notebbok.

Fabricio, E. (2014). Diseño e Implementación Del Prototipo de un Sistema Domótico para la Medición del Consumo de Agua Potable d Través de Internet y Correo Electrónico. Tesis. Licenciado en Ingeniería en Microelectrónica y Sistemas Distribuidos. Universidad Nacional de Loja. Colección de tesis digitales UNL.

Fundación de la Innovación - Bankinter. (2011). El internet de las cosas - En un mundo conectado de objetos inteligentes

Lorenzo, A. (2004). Comunicación Punto A Punto Vía Modem GSM. Tesis. Licenciado en Ingeniería Técnica Industrial en Electrónica Industrial. Universidad Rovira i Virgili. Colección de tesis digitales URV.

Miorandi, D. (2012). Internet of Things

Páez, M. (2013). Creación de un ambiente Tecnológico para el Diseño de Circuitos Integrados. Tesis. Licenciado en Ingeniería en Microelectrónica y Sistemas Distribuidos. Universidad de Los Andes de Venezuela. Colección de tesis digitales ULA.

Viveros, E. (2014). Sensor de Flujo de agua

Módulo Relay 1CH 5VDC. Disponible en <https://naylampmechatronics.com/drivers/297-modulo-relay-1-canal-5vdc.html> [accesado el 03 de mayo de 2019]

SIM900 GSM GPRS Shield Con Arduino Uno. Disponible en <https://www.instructables.com/id/SIM900-GSM-GPRS-SHIELD-CON-ARDUINO-UNO/> [accesado el 03 de mayo de 2019]

Soluciones 3D: Que es un Arduino. Disponible en <http://indesacompany.com/soluciones-3d/> [accesado el 03 de mayo de 2019]

ANEXOS

- **CODIGO DE CALIBRACIÓN DE SENSOR DE CAUDAL DE AGUA.**

```
/*CALIBRACIÓN DEL SENSOR DE CAUDAL DE AGUA YF-201*/
```

```
volatile int pulsos; // Cantidad de pulsos del sensor. Como se usa dentro de una interrupcion debe ser volatile
```

```
#define sensorDeFlujo 2 //Pin al que se conecta el sensor. DEBE ser 2 porque es unico que acepta interrupciones en el Arduino UNO
```

```
void flujo () // Funcion de interrupcion
```

```
{  
  pulsos++; // Simplemente sumar el numero de pulsos  
}
```

```
void setup()
```

```
{  
  pinMode(sensorDeFlujo, INPUT);  
  Serial.begin(9600);  
  attachInterrupt(0, flujo, RISING);  
  interrupts();  
}
```

```
void loop ()
```

```
{  
  Serial.print("Pulsos: ");  
  Serial.println(pulsos);  
}
```

- **CODIGO CALCULO DE CONSUMO DE AGUA POTABLE Y ENVIO DE MENSAJES.**

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial SIM900(7, 8); // Configura el puerto serial para el SIM900.
```

```
int respuesta;
```

```
char aux_str[50];
```

```
//Contenido del sms que enviamos. \x1A corresponde al caracter de finalizacion
```

```
char smsa[] = "LLAVE ABIERTA \x1A \r\n";
```

```
char smsc[] = "LLAVE CERRADA \x1A \r\n";
```

```
char mcubicos[]=" m3";
```

```
char smsprint[]="\x1A \r\n";
```

```
char convertirchar[50];
```

```

char concatenar[80];
char mensaje_recibido=0;
int relay=12;
int pos_memoria= 0;
volatile int pulsos; // Cantidad de pulsos del sensor. Como se usa dentro de
una interrupcion debe ser volatile
unsigned int litrosPorHora; // Calculated litres/hour
unsigned char sensorDeFlujo = 2; // Pin al que esta conectado el sensor
unsigned long tiempoAnterior; // Para calcular el tiempo
unsigned long pulsosAcumulados; // Pulsos acumulados
float metrosCubicos; // Metros cubicos acumulados

void flujo () // Funcion de interrupcion
{
    pulsos++; // Simplemente sumar el numero de pulsos
}

void calcular()
{
    if( millis() - tiempoAnterior > 1000)
    {
        tiempoAnterior = millis(); // Updates cloopTime
        pulsosAcumulados += pulsos;
        litrosPorHora = (pulsos * 60 / 6.30); // (Pulso de frecuencia x 60 min) /
7.5Q = flow rate in L/hour
        pulsos = 0; // Reset Counter
        metrosCubicos = ((pulsosAcumulados*1.00/378)/1000); //Cada 378
pulsos = 1 litro
    }
}

void setup()
{
    SIM900.begin(19200); //Configura velocidad del puerto serie para el
SIM900
    Serial.begin(19200); //Configura velocidad del puerto serie del Arduino
    pinMode(sensorDeFlujo, INPUT);
    pinMode(relay,OUTPUT);
    digitalWrite(relay, HIGH);
    attachInterrupt(0, flujo, RISING); // Interrupción para los pulsos
    interrupts(); // Habilitar interrupciones
    tiempoAnterior = millis();
    Serial.println("Iniciando...");
    power_on();
    iniciar();
}

void loop()
{
    calcular();
}

```



```

if (SIM900.available())
{
    mensaje_recibido=SIM900.read();
    switch (mensaje_recibido)
    {
        case 'A':// Si el mensaje es 'A' procede a abrir electroválvula
            digitalWrite(relay, HIGH);
            mensaje_sms(smsa);// Envía mensaje "LLAVE ABIERTA"
            break;
        case 'E'://Si el mensaje es 'E' procede a cerrar electroválvula
            digitalWrite(relay, LOW);
            mensaje_sms(smse);//Envía mensaje "LLAVE CERRADA"
            break;
        case 'R'://Si el mensaje es 'R' envía el consumo de agua
            dtostrf(metroscubicos, 6, 3, convertirchar);//Conversión del volumen a
char
            sprintf(concatenar,"%s %s
%s",convertirchar,mcubicos,smsprint);//concatenación
            mensaje_sms(concatenar);//envía mensaje con el volumen consumido
            break;
        }
    }
}

```

```

int enviarAT(char* ATcommand, char* resp_correcta, unsigned int tiempo)
{ int x = 0;
  bool correcto = 0;
  char respuesta[100];
  unsigned long anterior;
  memset(respuesta, '\0', 100); // Inicializa el string
  delay(100);
  while ( SIM900.available() > 0) SIM900.read(); // Limpia el buffer de
entrada
  SIM900.println(ATcommand); // Envía el comando AT
  x = 0;
  anterior = millis();// Espera una respuesta
  do { // si hay datos el buffer de entrada del UART lee y comprueba la
respuesta
    if (SIM900.available() != 0)
    { //Comprueba que no haya desbordamiento en la capacidad del buffer
      if (x < 99) {
        respuesta[x] = SIM900.read();
        x++;}
      else Serial.println("Desbordamiento!");
      // Comprueba si la respuesta del modulo es la 1
      if (strstr(respuesta, resp_correcta) != NULL)
      {
        correcto = 1;}
    }
  } // Espera hasta tener una respuesta

```

```

while ((correcto == 0) && ((millis() - anterior) < tiempo));
Serial.println(respuesta);
return correcto;
}

void power_on()
{
  int respuesta = 0;
  // Comprueba que el modulo SIM900 esta arrancado
  if (enviarAT("AT", "OK", 2000) == 0) //comprueba la conexion a la red
  {
    Serial.println("Encendiendo el GPRS...");
    pinMode(9, OUTPUT);
    digitalWrite(9, HIGH);
    delay(1000);
    digitalWrite(9, LOW);
    delay(1000);
    // Espera la respuesta del modulo SIM900
    while (respuesta == 0) {
      // Envia un comando AT cada 2 segundos y espera la respuesta
      respuesta = enviarAT("AT", "OK", 2000);
      SIM900.println(respuesta);
    }
  }
}

void power_off()
{
  digitalWrite(9, HIGH);
  delay(1000);
  digitalWrite(9, LOW);
  delay(1000);
}

void reiniciar()
{
  Serial.println("Reiniciando...");
  power_off();
  delay (5000);
  power_on();
}

void iniciar()
{
  enviarAT("AT+CPIN=\"1867\"", "OK", 1000);
  Serial.println("Conectando a la red...");
  delay (5000);

  //espera hasta estar conectado a la red movil
  while ( enviarAT("AT+CREG?", "+CREG: 0,1", 1000) == 0 )
  { }
  Serial.println("Conectado a la red.");
}

```

```

    enviarAT("AT+CLIP=1\r", "OK", 1000); // Activamos la identificacion de
    llamadas
    enviarAT("AT+CMGF=1\r", "OK", 1000); //Configura el modo texto para
    enviar o recibir mensajes
    enviarAT("AT+CNMI=2,2,0,0,0\r", "OK", 1000); //Configuramos el modulo
    para que nos muestre los SMS recibidos por comunicacion serie
    Serial.println("Preparado.");
}
void mensaje_sms(char* sms)
{
    if (enviarAT("AT+CREG?", "+CREG: 0,1", 1000) == 1) //comprueba la
    conexion a la red
    {
        Serial.println("Enviando SMS...");
        enviarAT("AT+CMGF=1\r", "OK", 1000); //Comando AT para mandar un
        SMS
        sprintf(aux_str, "AT+CMGS=\"962644717\"", strlen(sms)); //Numero al
        que vamos a enviar el mensaje
        //Texto del mensaje
        if (enviarAT(aux_str, ">", 100) == 1)
        {
            enviarAT(sms, "OK", 10000);
        }
        Serial.println("SMS enviado");
    }
    else
    {
        reiniciar();
        iniciar();
    }
}
}

```